# ST

MEI/JUNI 1989
nummer 19

PRIJS F 6,95 / BFR 143

ONAFHANKELIJK TIJDSCHRIFT VAN EN VOOR GEBRUIKERS VAN ATARI ST COMPUTERS




## Nieuwe Spreadsheets

## GfA Basic en Omikron Basic

## G + PLUS

## Modula-2: Megamax, SPC èn PD




ST 19 - voor ATARI ST COMPUTERS

ISSN 0923-2214

# Expert software®
## dé standaard voor professioneel automatiseren

De meest complete en betrouwbare programma's voor Atari ST computers. Of het nu gaat om de volledige automatisering van de administratie of alleen om de automatisering van de boekhouding, met Expert software® bent u gegarandeerd beter af. De grote voordelen van Expert software® zijn:

- Ondersteuning door een professionele organisatie,
- Volledige uitwisselbaarheid van alle Expert programma's,
- Veelzijdig en uiterst compleet,
- Doorgroeimogelijkheden,
- Volledige uitwisselbaarheid van Expert Junior, Boekhouding en Spreadsheet met 1ST Word Plus.

Het **Expert** leveringsprogramma omvat:

| | | |
|---|---|---|
| **Expert Junior:** | De boekhouding voor de kleine ondernemer of de starter | ƒ 595,-- en ƒ 795,-- |
| **Expert Boekhouding:** | De meest complete boekhouding voor het grotere bedrijf en de doorgroeier | ƒ 1495,-- en ƒ 1975,-- |
| **Expert Kostenplaatsen:** | Flexibele verwerking via de Spreadsheet | ƒ 250,-- |
| **Expert Facturering:** | Met macro's, artikellijsten en variabele layouts | ƒ 495,-- tot ƒ 1195,-- |
| **Expert Voorraad:** | Voor doelmatig voorraadbeheer | ƒ 325,-- tot ƒ 1275,-- |
| **Expert Spreadsheet:** | Superieure spreadsheet voor o.a. voor- en nacalculatie, budgettering, statistiek en orderadministratie, razendsnel | ƒ 540,-- |
| **Expert Salaris:** | Uiterst compleet, terugboeken het gehele jaar mogelijk | ƒ 1975,-- |

In alle Expert programma's standaard: calculator, formuleprocessor en macro's. Maatwerkapplicaties tegen meerprijs.

Inlichtingen: bij MEGAsystems of één van de geautoriseerde Expert software®-dealers




## Softwaretrainingen

Dagcursussen tekstverwerking, desktop publishing en database-management voor de zakelijke gebruiker met Atari ST-computers. Nu kunt u snel op een efficiënte manier vertrouwd raken met de Atari ST-computerlijn en de gangbare softwarepakketten. Deelname is mogelijk na inschrijving en na vooruitbetaling van de deelnamekosten. De cursusdata worden in overleg met de deelnemers bepaald. Bij de cursus zijn inbegrepen: het gebruik van Atari-computerapparatuur, schriftelijk cursusmateriaal, lunch en consumpties. De kosten per cursus zijn ƒ 255,--. Vraag uw dealer naar een inschrijfformulier of bel MEGAsystems.

Alle prijzen excl. BTW.

**MEGAsystems**

Doelenstraat 14 · 6711 AR EDE ☎ 08380-10010

ABONNEMENT & LOSSE NUMMERS


ST is een onafhankelijk tijdschrift van en voor gebruikers van Atari ST computers.

**ISSN 0923-2214**

**Uitgever en redactieadres**
Stichting ST
Bakkersteeg 9a
2311 RH Leiden
postadres: Postbus 11129
2301 EC Leiden
telefoon: 071-130045

**Redactie ST**
Bert Rozenberg: hoofdredacteur
Peter Hendriks: coördinator
Han Driesen: eindredacteur

**Redactie-medewerkers**
Berg Chabot
Vincent van Diemen
Ronald J. van der Kamp (Redactiesecretaris en Stichting Modula)
Wil Kerkhof
Johan Lammens (contactpersoon België)
Hans van Oudenaarden (illustraties)
Guus Ramackers (Stichting MicroMusic)
Egbert de Rooij
Jos Smeets
Eke van Batenburg
Tom den Duyf
Erik van Straten

**Verder werkten aan dit nummer mee**
Sander Vriesman
Mr. V.A. de Pous
Wouter Schuurmans
Jan Willem van der Veen
Hans Vermeulen
Herman de Vrees

**Foto's voorpagina:**
Victor Bestebroer

**Abonnementen**
Het tijdschrift ST verschijnt zes maal per jaar. Een abonnement voor zes nummers kost fl 35,-. Abonnementsgeld kan worden overgemaakt op postgirorekening 1626826 tnv. Stichting ST, Bakkersteeg 9a, 2311 RH Leiden.
Vragen over abonnementen kunt u stellen via telefoon: 071-130045 (9.00-10.30 en 18.30-19.30 op werkdagen).
Nadere informatie is te vinden op de servicepagina van de Stichting ST in dit blad.

**Advertentiewerving**
Bosker & Noordhuizen
Mediaservices
Postbus 22
2302 AA Leiden
telefoon:
071-123975
071-123674
(tarievenkaart op aanvraag)

**Distributie**
Betapress B.V., Gilze
Telefoon: 01615-7800

**Druk en vormgeving**
Het Urkerland B.V.
Telefoon: 05277-4455




# INHOUDSOPGAVE



# ADVERTEERDERSINDEX

# Redactioneel

ST was op de Hannover Messe vertegenwoordigd door twee medewerkers. In een uitgebreid verslag wordt een overzicht geboden van een groot aantal nieuwe (of bijna nieuwe) zaken.
Het inleidende overzicht van het OS-9-systeem wordt dit keer afgerond; voor velen een leerzame eerste rondgang door een multi-tasking/multi-user omgeving.
Het mag wel weer eens benadrukt worden dat het uitbrengen van het ST-blad alleen mogelijk is dankzij de belangeloze inzet van alle medewerkers, voor wie dit alles het spreekwoordelijke 'liefdewerk en oud papier' is. Juist de niet-commerciële basis waarop wij met elkaar ST maken, vereist een subtiel samenspel van alle betrokkenen.

Misschien zoudt u graag een bepaald onderwerp besproken zien in dit blad; laat het ons a.u.b. weten en wij zullen proberen een deskundig auteur hiervoor te strikken. Ook wij hebben niet altijd de juiste 'neus' voor wat er zoal belangwekkend is voor onze lezers.
Sommige artikelen zijn misschien niet voor iedereen interessant. ST wil echter ook een forum bieden voor onderwerpen die ergens anders moeilijk aan bod komen. Als uw specialisme of interessegebied ontbreekt, schroom dan niet om zelf een bijdrage te leveren. Wellicht kunt u anderen inspireren. Bedenk dat veel ST-bezitters de platgetreden paden willen vermijden, anders hadden ze wel een conventionele machine gekozen.

Ook deze keer willen we een van onze nieuwe medewerkers bij u introduceren: Sander Vriesman. Hij is veruit onze jongste medewerker. Als 'games'-specialist bij uitstek zal hij in de toekomst vast nog veel bijdragen in de rubriek ST GAMES publiceren. Wij kennen geen leeftijdgrenzen voor onze medewerkers. U bent nooit te jong of te oud.

Zoals u wellicht in het vorige nummer van ST al heeft gezien, zijn sommige namen van vaste rubrieken gewijzigd. Met deze nieuwe namen hopen we de herkenbaarheid van deze rubrieken te vergroten. Wij proberen steeds verbeteringen door te voeren. Het is onze bedoeling dat ST het onafhankelijke tijdschrift van en voor gebruikers van Atari ST computers blijft.

(ph/hd)



# Agenda

## Nederlandse gebruikersgroepen

Gebruikersgroep **Almere**
Dhr. Keyzer (03240-39229).

Gebruikersgroep **Amsterdam**
Bijeenkomsten iedere eerste maandag van de maand in gebouw 'De Meerpaal', van Woubruggestraat 50 in Amsterdam om 19.30 uur.
T. van Rooy (020-164562).

SAG **Arnhem-Nijmegen**
Bijeenkomsten iedere eerste zaterdag van de maand vanaf 13.30 uur in het gebouw 'De Kinkel', Dorpsstraat 1b in Bemmel.
Kees Verhaaf (08365-1510).

**Bredase**
Regionale Atari Computerclub (BRAC)
Bijeenkomsten elke eerste en derde donderdag van de maand van 20.00 tot 23.00 uur in gemeenschapshuis 'Dorenbos', Abdijstraat 26 in Breda.
Erik Mareels (01613-1319).

Gebruikersgroep **Delft**
Bijeenkomsten elke eerste donderdag van de maand vanaf 19.00 uur in 'Hotel Central', Wijnhaven 6 in Delft.
A. Balk (015-142397).

Atari Hobbyclub **Den Helder**
Geen specifieke ST gebruikersgroep, wel een groeiend aantal ST- bezitters onder de leden. Bijeenkomsten in principe elke laatste vrijdag van de maand in 'Sint Nicolaas Centrale', Loodsgracht 55, Den Helder (02230-18075).
P. de Leeuw, Arie Kieboomstraat 64, 1785 HL Den Helder.

SAG **Doetinchem** en omgeving
Marcel Prinsen (08340-41274).

Gebruikersgroep **Dordrecht**
Bijeenkomsten elke tweede dinsdag en vierde woensdag van de maand van 19.30 tot 23.00 uur in buurthuis "t Wanty', Maasstraat 23 in Dordrecht.
J. Schreuders (078-148700).

STichting Computer **Eindhoven**
Gebruikersbijeenkomsten elke eerste zaterdag van de maand (10.30-15.30) in 'De Ketting', Tinelstraat 3a in Eindhoven. Verder een dinsdagavond per maand in de Evenementenhal te Eindhoven. Er is een eigen tijdschrift 'CLIPBOARD'.
Jos Smeets (04902-40032).

Sag regio **Eemland**
Elke tweede dinsdag van de maand een bijeenkomst in 'De Til', Hamersveldseweg 30 te Leusden. Aanvang 19.30 uur.
Martin Nefkens (033-945026).

Atari gebruikersgroep **Friesland**
Iedere maand een bijeenkomst in het clubgebouw van de Orca. AGGF, postbus 72, 8900 AB Leeuwarden.

**Gooise** Atari Gebruikers (G.A.G)
Elke eerste zaterdag van de maand bijeenkomsten in de 'Malbak', Wetering 120, Blaricum-Bijvanck.
Tom Demetrius (035-15660).

Atari Club **Groningen** (A.C.G.)
Bijeenkomsten om de veertien dagen in wijkcentrum 'de Beijumkorff'.
Henk Postema (050-414236).

Stichting **GFA** gebruikersgroep
Geen bijeenkomsten, wel het eigen blad: GFA-Info Inlichtingen: redactie GFA-Info, Frankrijklaan 14, 2034 BA Haarlem of telefonisch (023-336946).

**HCC Atari** gebruikersgroep
Bijeenkomsten van 10.30 tot 16.00 uur in Technische School 'De Bron', Vaderreindreef 7 in Utrecht (bij NS- station Utrecht-Overvecht). In 1988 nog op 10 september en 3 december.
Frits Zwart (020-793359).

Gebruikersgroep **Heerhugowaard**
Elke eerste zaterdag van de maand een bijeenkomst vanaf 13.30 uur in het Olympiagebouw, Olympiaweg 17, Alkmaar.
Gerard Knaven (02202-1663).

Computerclub **Hoogeveen**
Bijeenkomsten van de MSX en Spectrum gebruikersgroepen samen met de Stichting AGNON (Atari Gebruikers Noord en Oost Nederland) vanaf 19.30 uur in 'De Tamboer', Hoogeveen.
O. Roorda (05280-69366).

Gebruikersgroep **Hoorn**
Iedere tweede zaterdag van de maand een

bijeenkomst van 14.30 tot 16.30 uur in de Kerkzaal van 'De grote Beer' in Hoorn.
Inlichtingen: 02290-38809.

ST/68000 groep **Leiden**
Bijeenkomsten elke eerste donderdag van de maand van 20.00 tot 22.15 uur in het Centraal Reken Instituut aan de Wassenaarseweg 80 in Leiden.
Tom den Duijf (071-134371).

SAG **Limburg**
Elke eerste maandag van de maand een bijeenkomst in 'Terwinselen', Schaersbergerweg 27, Kerkrade-West van 19.30 tot 22.30 uur.
Wil Braakman (045-418695).

Regio **Nieuwegein**
Bijeenkomsten om de twee weken op de dinsdagavonden,in buurthuis 'De Bongert', Moerbeigaarde, Nieuwegein.
S. Hamstra (033-940611).

ST **Oost**
Bijeenkomsten iedere eerste dinsdagavond van de maand in 'Twente Technovia', Walstraat 2, Enschede. Aanvang 20.00 uur.
Henno Tillema (053-331876).

Atari club regio **Oss**
Wekelijkse bijeenkomsten 's maandags vanaf 19.00 uur in wijkcentrum 'De Hille', Looveltlaan 3, Wijk Ussen, Oss.
[illegible] de Jong (04120-36036).

[illegible]egio **Rynmond**
[illegible]komsten elke tweede zaterdag (overdag) [illegible]e maand in buurthuis 'De Kreek' in De [illegible]kers, Spijkenisse, elke eerste maandagavond [illegible]n de maand in buurthuis 'Ricardo' aan de Arend van de Woudenslaan in Lombardije en elke derde donderdag van de maand in wijkhuis 'Jaffa', Weteringstr. 275 Rotterdam Kralingen.
Ron Koolen (010-4554848) en Henny Hopman (010-4803510).

ST Club **Spaubeek-Limburg**
Elke tweede zaterdag van de maand een bijeenkomst van 11.00 tot 16.00 uur in het 'Multifunctioneel Centrum' in Spaubeek.
M. Claessens (04490-52532).

ST gg **Tilburg**
Bijeenkomsten elke derde woensdag van de maand in wijkgebouw 'De Schans', Schans 123, Tilburg.
Jos Moonen (03166-2617) of G. de Beer (013-550617)

Stichting Computerclub **Veldhoven**
Iedere laatste zaterdag van de maand een bijeenkomst van 11.00 tot 16.00 uur in "t Patronaat' aan de Blaarthemseweg 18 in Veldhoven.
Engel Geukens (040-535099).

Atari Computerclub **Veluwe/IJsselstreek**
Bijeenkomsten om de veertien dagen op dinsdagavond van 19.30 tot 22.15 in Sportcentrum 'de Maten', Ambachtsveld 2 te Apeldoorn.
T.B. Hamstra 055-420179 of 415242.

Atari ST-club **Venlo**
Iedere derde zaterdag van de maand een bijeenkomst van 10.00 tot 12.00 in klubgebouw 'De Viking', Vikingweg te Venlo.
Wibo Visser (077-512568), Leutherweg 51 A, 5915 CB Venlo.

ST-Club **Wageningen**
Gebruikersbijeenkomsten elke derde woensdag van de maand vanaf 20.00 uur in 'de Leeuwenborch', Hollandseweg 1 in Wageningen. Maandelijks het clubblad ST-FOLDER met tips en wetenswaardigheden.
Jack Schoenmakers (08370-18258)

ST-gebruikersgroep **Zeeland**(STZ)
Bijeenkomsten in buurthuis 'De Pit' te Goes op elke tweede dinsdag van de maand vanaf 20.00 uur.
R.Bitter (01100-23689) of C.Jansen (01192-2195)

Atari gg **Zoetermeer** (Z.A.G.G.)
Bijeenkomsten op de tweede donderdag van de maand in wijkcentrum 'Buytenwegh', Teldersrode 25, Zoetermeer.
G. Verbaan, Vissendreef 43, Zoetermeer.

Atari ST **Zwolle**
Elke tweede vrijdag van de maand vanaf 20.00 uur een bijeenkomst in wijkcentrum 'Zwolle-Zuid', Munterkamp 75, Zwolle.
Ed. Hoogkamer (038-652859).

## Vlaamse gebruikersgroepen

GAST, **Aalst**
Mark Barbe (053-773399).

ACB, **Antwerpen**
zie Triple A, Antwerpen

HCC Atari ST, **Antwerpen**
Patrick Op de Beeck (03-4494445).
Vergadering elke eerste woensdag van de maand op de eerste verdiepindg van de HCC-lokalen, Kleine Markt 7-9 te Antwerpen

Triple A, **Antwerpen**
bijeenkomsten in 'de Nieuwe Carnot', Carnotstr 60, Antwerpen op dinsdag en donderdagavond.
Koen van der Elst (03-2329855).

Stichting ST, **België**
Johan Lammens (091-231624).

VTB-VAB Atari gebruikersgroep **Deinze**
Elke derde zondag van de maand van 9.00 tot 12.00 uur in lokaal 'Palaestra' te Deinze.
Dominique Fierens (091-868689).
Kontaktman in Nederland: Stan van Zon (071-173638).

WESTRAND, **Dilbeel**
Luc Mertens (02-5693402).

VTB-Atari Computerclub **Evergem**
VTB-lokaal, Hekstraat 7, B 9050 Evergem, België. Tweewekelijkse vergaderingen; eenmaal op dinsdagavond (19.00-23.00 uur) en eenmaal op zondag (10.00-17.00 uur).
Patrick Delabarre (091-262929 na 19.00 uur).

Floppy ACC **Genk**
Elke tweede zaterdag van de maand in het gemeentehuis te Genk.
Frans Liekens, (011-354893).

Atari ST club **Gent**
Maandelijkse vergaderingen op maandag om 20.00 uur in café 'Willen is Kunnen' in de St. Coletastraat in Gent.
Filip Nevejans, (091-224329).

VAST, **Gent**
Eric Blanpain (091-301954).

CAST, **Hasselt**
Mark Broucks (011-221394).

HCC Atari ST, **Kortrijk**
Frans Van De Maele (056-667548).

ISTARI Workshop, **Mechelen**
Gust Vervoort (015-211020).

HCC België Modula-2 Interesse Groep
Patrick Op de Beeck (03-4494445)

## Internationaal

ST Club **International**
Een internationale ST- club zonder bijeenkomsten, maar met het PD Disk- magazine ST BULLETIN. Henk Boer, ST Club International, Postbus 13672, 2501 ER 's-Gravenhage, Nederland.

De redactie van ST zoekt contact met de in deze agenda niet opgenomen ST-gebruikersgroepen. Vooral onze Vlaamse collega's worden hiertoe uitgenodigd. Correspondentie o.v.v. 'Agenda' richten aan:
**Stichting ST,**
Bakkersteeg 9a,
2311 RH Leiden.



# Vragen HOEK

Na de 520 ST, de 1040 ST, de Mega ST2 en de Mega ST4 is er inmiddels ook een Mega ST1. Of dit er toe zal leiden dat de ST meer verkocht gaat worden, valt nog te bezien. Waarschijnlijk zullen de prijzen van de ST binnenkort wel dalen. Dit zal er voor zorgen dat de ST voor (nog) meer mensen bereikbaar wordt. Mensen die waarschijnlijk nog nooit een computer hebben gehad. Atari zal zijn ondersteuning daaraan moeten aanpassen. Hoe dan ook, vragen kunt u altijd aan ons kwijt. Ook tips die u heeft voor andere gebruikers stellen we bijzonder op prijs.

Uw vragen, tips en opmerkingen kunt u op de volgende wijzen kwijt:

Per post:
Stichting ST
Postbus 650
3200 GA Spijkenisse

Electronisch:
STER BBS
01880-40035 (V21, V22, V22-bis)
01880-42472 (V23)

Telefonisch:
Alle dagen van 20.30 tot 21.30 uur !!
01880-33083

**Vraag:**
Ik heb een zelfde printer als een collega van mij. Toch krijg ik met WordPlus een andere uitvoer op papier. Hoe kan dit?

**Antwoord:**
Er zijn twee mogelijkheden waardoor dit kan. Ten eerste maakt WordPlus bij het printen gebruik van een printer-driver. In deze printer-driver staan de codes die WordPlus naar de printer stuurt om bijvoorbeeld het onderstrepen aan te zetten. Als uw collega een andere printer-driver bezit dan u, dan zal de uitvoer naar de printer ook anders zijn. Een tweede mogelijkheid is de instelling van de printer zelf. Bijna elke printer bezit een aantal dipswitches. Dit zijn hele kleine schakelaartjes waarmee u bepaalde instellingen van uw printer kunt veranderen. Zo kunt u bijvoorbeeld instellen of de lengte van uw papier 11 inch of 12 inch is. Veel printers bezitten een dipswitch waarmee de printer verteld kan worden of deze zich moet gedragen als een Epson- of als een IBM-printer. Buiten de verschillende karaktersets vormen het grootste verschil de commando's voor grafische afdrukken. De ST geeft bijvoorbeeld een schermafdruk als u op ALTERNATE-HELP drukt. Deze grafische afdruk gaat volgens de Epson-standaard. Mocht uw printer niet netjes het scherm afdrukken, dan is het mogelijk dat deze ingesteld staat op de IBM-emulatie of zich niet netjes als een Epson gedraagt. In het eerste geval kunt u de instelling veranderen. In het tweede geval is er niet veel te doen. In de Public-Domain is echter wel een programma waarmee u Degas- printer-drivers in kunt laden, welke dan de screendump-functie overnemen.

**Vraag:**
Is het mogelijk om 2 ST's aan te sluiten op 1 harddisk of 1 printer?

**Antwoord:**
Het aansluiten van 2 ST's op 1 harddisk is wel mogelijk. Een soort mini-netwerkje is wel te maken. Er zijn zelfs wat Public-Domain programma's die een netwerkje maken door middel van wat kabels aan de midi-poort. Echt betrouwbaar werkt dit echter niet. Daarnaast waren er op de Cebit (de beurs in Duitsland) wel wat netwerkjes te zien, waarvan minstens 1 in ieder geval betaalbaar. Over de betrouwbaarheid valt echter niets te zeggen. Vaak is het echter goedkoper om een tweede harddisk aan te schaffen. Zeker de aangekondigde harddisks met verwisselbare schijven zijn bijzonder interessant. Het aansluiten van 2 ST's op 1 printer is geen enkel probleem. Voor een paar tientjes koopt u een schakelkastje waarmee de printer d.m.v. een knop van de ene ST naar de ander geschakeld kan worden. Zo'n kastje is echter door de gebruiker ook in een uurtje in elkaar gezet. Voor bedragen van 200 gulden en meer zijn intelligente printerschakelaars te koop. Deze bevatten dan ook vaak een printerbuffer.

**Vraag:**
Ik heb een 5 1/4-inch-diskdrive op mijn ST aangesloten. Deze is voorzien van een 40/80-tracks-schakelaar. Als ik nu diskettes mee naar mijn werk neem, dan werkt dit prima, maar op mijn eigen PC kan ik de diskettes niet lezen. Op mijn werk hebben we een 80-tracks drive en mijn PC heeft een 40-tracks-drive.

**Antwoord:**
Als we 80 tracks op een diskette willen schrijven, dan dienen deze tracks uiteraard smaller te zijn dan wanneer we er maar 40 willen schrijven. De tracks die een 80-tracks-drive schrijft, zijn dus smaller dan de tracks van een 40-tracks-drive. Zelfs als we met de 80-tracks-drive maar 40 tracks beschrijven. Nu komt het regelmatig voor dat de tracks die geschreven zijn met een 80-tracks-drive te smal zijn om door een 40-tracks-drive gelezen te kunnen worden. Het enige wat wel eens wil helpen is de diskette te formatteren op de 40 tracks drive.

**Bert Rozenberg**

# Public Domain-Software: copiëren toegestaan.

**MR. V.A. DE POUS**

**Over de mogelijkheden en onmogelijkheden voor juridische bescherming van computerprogrammatuur in Nederland heerst soms nog onduidelijkheid, ondanks het feit dat de kaarten reeds geschud zijn.**

Even een paar vuistregels. Het Nederlandse auteursrecht is geregeld in de Auteurswet 1912, aangevuld met flink wat rechterlijke uitspraken. Ons auteursrechtsysteem is open, wat wil zeggen: in beginsel kan van alles en nog wat onder de werking van dit intellectueel eigendomsrecht worden gebracht. Het auteursrecht ontstaat op moment van schepping van het werk (registratie is geen voorwaarde), indien het idee dat er aan ten grondslag ligt voldoende is uitgewerkt en indien de uitwerking ervan van enige originaliteit getuigt. Men spreekt in dit kader wel van statistische creativiteit. Leiden er meerdere wegen naar Rome of zijn bijvoorbeeld technische randvoorwaarden dusdanig van aard dat slechts één route kan worden gekozen om het beoogde resultaat te verkrijgen? Deze vragen staan bij auteursrechten centraal. Ook in het perspectief van software.

Het auteursrecht zelf is het uitsluitende recht van de maker zijn werk te openbaren en te verveelvoudigen. Behalve dit exploitatierecht kent het auteursrecht ook nog het zogenoemde persoonlijkheidsrecht van de maker, dat in tegenstelling tot het exploitatierecht niet overdraagbaar is. Verkoopt Bill Gates de rechten op Microsoft Word, dan mag de nieuwe rechthebbende niet zonder toestemming van Gates het programma aanpassen. Zo simpel is dat.

De Nederlandse automatiseringsbranche heeft zich in het recente verleden met succes sterk gemaakt voor aanpassing van onze auteurswetgeving. Het begrip 'computerprogramma's' zal uiteindelijk later in de wet worden opgenomen, terwijl het softwaregebruikers onder meer - regelend recht - wettelijk wordt toegestaan een backup-copie van de software te maken. Wel gaan de maximum straffen bij piraterij omhoog en ook afdracht van de genoten winst wordt straks voor het slachtoffer mogelijk. Vele rechterlijke uitspraken laten echter zien dat onze huidige wet van voldoende flexibiliteit getuigt om computerprogramma's in source-of object-code, op schijf of tape en in RAM of (E)(P)ROM juridisch te beschermen. En zelfs electronische schakelingen vallen onder de werking van de uit 1912 stammende Auteurswet. Hetzelfde geldt voor chips die voor 7 november 1987 zijn gemaakt. Op topografieën van halfgeleiderprodukten die op of na die datum zijn ontworpen is echter de nieuwe Topografiewet van toepassing. Van de komende wijziging in de Auteurswet zal dan ook naar alle waarschijnlijkheid slechts een preventiewerking uit gaan. Feitelijk verandert er weinig. Dateert de discussie op de Nederlandse softwaremarkt over rechtsbescherming van computerprogramma's van ongeveer 1980, van jongere datum is de vraag naar de juridische status van public domain-software. Hierbij gaat het om programmatuur, waarvan de rechthebbende afstand van zijn rechten heeft gedaan of waarbij hij toestemming heeft gegeven tot de mogelijkheid van copiëren ervan, zonder dat hiervoor door de gebruikers betaald moet worden. Deze toestemming kan weliswaar onder bepaalde voorwaarden worden verleend, maar als programmatuur eenmaal in het public domain-circuit terecht is gekomen, blijft van de eventuele nadere bepalingen weinig over. Auteursrechtelijk beschermde werken komen echter ook in het public domain terecht als het auteursrecht is afgelopen en dat betekent in beginsel vijftig jaar na de dood van de auteur.

De hoofdregel voor public domain-software luidt: eenmaal (weg)gegeven blijft (weg)gegeven. Rechtmatige public domain-software die in Nederland in omloop is, kan niet ineens tot programmatuur verworden waarvan de auteursrechten wel volledig worden uitgeoefend. Gebruikers mogen erop vertrouwen dat rechtmatige public domain-produkten in het public domain blijven. Tevens is van belang wat het computerprogramma zelf zegt over haar status. Verschijnt er op het scherm een copyright, wordt er aangegeven welke handelingen toegestaan dan wel verboden zijn? Dit zijn zaken die voor de gebruiker van belang zijn.

Juridische problemen rond public domain-software zijn meestal van één soort: de status van het softwareprodukt is niet duidelijk, waardoor er illegaal wordt gecopieerd. Illegaal wil zeggen dat verveelvoudiging zonder toestemming van de rechthebbende plaatsvindt. Er zijn geen rechten betaald.

Maar wat gebeurt er als de public domain-exploitant een nieuwe versie van zijn softwareprodukt als 'gewone' programmatuur op de markt brengt? Als rechthebbende heeft hij deze bevoegdheid, maar de leverancier moet dan wel in de software en op de verpakking aangeven dat het programma auteursrechtelijk beschermd is. Daarnaast zal in Nederland de redelijkheid gebieden dat de veranderde bedrijfsfilosofie bijvoorbeeld door middel van advertenties in computer(hobby)bladen ter kennis van gebruikers en potentiële gebruikers wordt gebracht. Ook is het waarschijnlijk dat de nieuwe versie van de software aanzienlijke veranderingen ten opzichte van de vorige, gratis versie heeft.

Shareware is echter wat anders dan PD-software. Daar rusten wel auteursrechten op. Maar daarover en over de zaak ButtonWare vs. HCC een andere keer.

**Mr. V.A. de Pous.**

*Mr. V.A. de Pous houdt zich bezig met recht en informatieverzorging inzake juridische aspecten van de informatietechnologie en is onder meer uitgever/redacteur van de maandelijkse nieuwsbrief NewsWare en juridisch medewerker van het automatiseringsvakblad Computable.*

# ST actueel

## Megafile 44

Als men dit leest is het misschien al mogelijk om in de winkel een verwisselbare harddisk van 44 Mbyte te kopen van het merk Atari. Deze drive is aan te sluiten op ST's. De verwachting is dat men een paar duizend gulden ervoor moet uitgeven. Er zal een nieuwe AHDI komen, die in staat zal zijn om met verwisselbare media te werken. Er zal ook een CD-ROM-drive beschikbaar komen, en in de toekomst zal er een revisie komen van het operating systeem. Er wordt gezegd dat er een MetaDos in de maak is, die het mogelijk zal maken om allerlei soorten drivers voor diverse apparaten te installeren. Afwachten maar.

## Postscript

Voor bonafide software-ontwikkelaars is er via Atari (Duitsland) de zgn. Ultrascript GEM-applicatie te verkrijgen. Met minimaal een Mega 2 en een SM804 laserprinter, is het mogelijk om applicaties te ontwerpen die Postcript uitvoer leveren.

## Enigma

Gedurende de tweede wereldoorlog werden door het Duitse leger gecodeerde berichten gemaakt en ontcijferd met de Enigma-machine. Momenteel is er een softwarematig gemaakte Enigma te verkrijgen, die op de ST draait. Voor DM 49,- bij: SGG System-Entwicklung, Mainhardingstr. 30. D-4402 Greven, Duitsland.

## Is GDOS er?

Als men een programma gaat draaien dat GDOS nodig heeft, en GDOS is niet geïnstalleerd, dan zal de boel instorten met veel bommen op het scherm. Vooral een aanroep van 'Load Fonts' weet in een dergelijke situatie een spectaculair aantal bommen te produceren. Voor mensen die zelf programma's maken, is er altijd te mogelijkheid om te testen of GDOS er is en wel op de volgende manier:

```
MOVE.W # -1,D0
TRAP # 2
CMP.W # -2,D0
SNE D0
EXT.W D0
```

Als er op die manier een 0 in register D0 staat, dan is GDOS niet aanwezig.

## Nogmaals SCART

In de vorige uitgave van ST is aandacht besteed aan het aansluiten van SCART TV's aan de ST. Toch wil er soms wel eens een TV zijn, die de SCART ingang afwijkend behandelt. Wat men ook doet, soms blijft men een flikkerend beeld houden. In dat soort gevallen valt het te proberen om de uitgang van pin 12 van de ST (de vertical sync) te verbinden met pin 16 van de SCART (het blanking signal). Voor mensen die heel voorzichtig met hun ST omspringen, kan eventueel een weerstandje van 2k2 Ohm in deze leiding opgenomen, wat extra beveiliging bieden.

## Smalltalk

Het Smalltalk-80 System Version 2.3, gemaakt door ParcPlace Systems Inc. in Palo Alto, California, is in Duitsland te krijgen. De versie voor de ST (benodigd: megaST en harddisk) is te krijgen voor ongeveer DM 2250,-. Nogal veel voor de meeste mensen, maar gelukkig is er een korting voor leraren, scholieren, studenten, zodat die maar een DM 400,- hoeven te betalen. Daarmee komt dit pakket ook binnen het bereik van de gewone ST-bezitter (immers een goede compiler voor C of Modula kost ook zoveel). Voor universiteiten en vergelijkbare instellingen is er ook een speciale licentieregeling.
De ST is nu de goedkoopste hardwarebasis waarop SmallTalk draait.
Inlichtingen: Georg Heeg, Baroperstr.337, D-4600 Dortmund 50, Duitsland, tel. 0231-751326.

## Floppydrives

De prijzen van direkt aan de ST aansluitbare, kant-en-klare drieëneenhalf inch diskdrives zijn nog verder aan het zakken. In Duitsland ligt de prijs voor een enkele drive op DM 240,- en voor twee drives in een kast op DM 470,-. De prijzen van vijfeneenkwart inch drives zijn ook verder gedaald. Een 40/80 track omschakelbare drive hoeft daar maar DM 400,- te kosten. Mij blijft het een raadsel waarom in ons land dit spul steevast honderd gulden duurder moet zijn.

## Signum

Deze bekende editor kan nu ook gebruikt worden voor de volgende talen: Arabisch, Bengaals, Sanskrit, Farsi, Grieks en Gujerati. Helaas nog geen Pali. Deze fonts, met utilities om van rechts naar links te typen, zijn verkrijgbaar voor 25 pond per stuk bij Gate Seven Computers (Mike Meir), 6 A Gwendwr Road, London W14 9BG, Engeland, tel. 01-6025186.

## Hardware en virus

Laatstelijk kwam de vraag binnen of het mogelijk was dat een virus een ST beschadigde. Er kan hierover het volgende gezegd worden:
1. Floppydrives. De motor zelf kan beschadigd worden als die gedurende langere tijd snel aan- en uitgeschakeld wordt. Ook kunnen door de hoge aanloopstroom de stuurtransistors doorbranden. Het geluid dat de drive produceert, zorgt er voor dat elke gebruiker snel zijn machine afzet. Niet zo'n probleem dus. Een andere mogelijkheid is de beweging van de lees/schrijfkoppen. Softwarematig kan er nogal gegoocheld worden met de floppycontrollerchip, zodat het mogelijk is om zelfs een drummachine van je drive te maken. Over het algemeen is het mechaniek best wel bestand tegen mishandeling: doorstappen naar b.v. track 100, die er niet is, geeft geen mechanische beschadiging. Wel bestaat de mogelijkheid dat er mechanische oscillaties ontstaan, maar het geproduceerde geluid zal een ieder snel zijn machine doen uitzetten.
2. De printerpoort. Het busy-signaal van de printer loopt naar de MFP-chip. Wordt deze chip anders geprogammeerd, dan is het mogelijk om dat lijntje niet meer als ingang maar als uitgang te configureren. Dan botsen twee uitgangssignalen op elkaar en kan er òf in de printer iets kapot gaan, òf de MFP-chip in de ST geeft de geest. Remedie: printer afzetten als die niet nodig is, speciaal in situaties waarin er besmettingsgevaar is geweest (of nog is).
3. De MMU. Deze chip is in CMOS-techniek uitgevoerd, en trekt alleen stroom (en produceert dan dus warmte) als hij bezig is. Gelukkig wordt de MMU normaal niet zo zwaar gebruikt dat hij doorbrandt, maar software kan natuurlijk altijd deze chip mishandelen, door hem voortdurend aan het werk te houden. Dan kan de boel oververhit raken en de geest geven.
4. De harddisk. Deze is altijd al een geliefd doel geweest van gore virussen. Het wissen van directories, FAT-tabellen e.d. door virussen heeft al heel wat mensen schade toegebracht. Het is bij sommige merken harddisks zelfs mogelijk om de zgn. servo- informatie te veranderen. Dan wordt de drive zelfs onbruikbaar, omdat de koppen niet gepositioneerd kunnen worden. Tot nu toe is is ons iets dergelijks niet

gemeld, want er moet heel wat gedaan worden wil men via de harddisk-controller dit effect kunnen bereiken. Directories wissen is veel simpeler.

(rjvdk)

## STEVE

Het geïntegreerde zakelijke Atari-programma STEVE (ST EVent Editor) is nu uiteindelijk ook in Nederland te koop in versie 3.23.
Dit programma is ontwikkeld door Primoz Jakopin uit Ljubljana. In minder dan een half Megabyte geheugen zitten: tekstverwerking, database, DTP, grafieken, font-editor, communicatie, scanner-driver, agenda, calculator en noem maar op. Dit programma is al eens in dit blad besproken (zie ST17). De prijs is slechts fl 380,- ex BTW, en de updates tot versie 4 zijn gratis. Voor beginners is er een Hot-Line.
De leverancier, die ook de support voor Nederland doet is: Theo Richel Producties, Stationsstr. 43, 4421 AK Kapelle, tel: 01102-44144.

## GG Antwerpen

In Belgie is een fusie tot stand gekomen tussen Triple A en ACB, beide uit Antwerpen. Hierbij wordt de nieuwe vereniging de grootste Atari gebruikersvereniging van België. Het eigen tijdschrift ROM zal worden uitgebreid. Men hoopt de dienstverlening aan de leden uit te kunnen breiden. Algemene bijeenkomsten zullen gehouden worden op elke eerste en derde dinsdag van de maand, de tweede dinsdag is er de MIDI-groep, de derde dinsdag de GFA-groep, op de eerste donderdag de GFA-beginnerscursus en op de [illegible] donderdag van de maand de GFA-[illegible]groep. Het correspondentie adres van de vereniging blijft: Otto Vebiusstr 26/32, 2000 Antwerpen, en de activiteiten vinden plaats in 'de Nieuwe Carnot', Carnotstr 60, Antwerpen.

## That's Analyse

De lancering van That's Analyse is uitgesteld tot september 1989. Het pakket is weliswaar al produktie-klaar maar Theo Breuer van Compo geeft aan, dat er nu even ruimte gemaakt moet worden voor de LDW Power van ATARI. Als er wat rust op de markt komt, dan is het nog tijd genoeg voor That's Analyse. Voor informatie: 04458-1797

## That's Text Publishing

Alweer een DTP-pakket, maar nu uit de That's-serie, moet uitkomen in september van dit jaar. Dit pakket kent een eigen tekst-editor, die natuurlijk That's Write compatible is. Het wordt voor elke geregistreerde That's Write gebruiker mogelijk, om zijn pakket in te leveren en voor een bescheiden bijbetaling That's Text Publishing ontvangt. In dit TTP-pakket zit een geïntegreerd tekenpakket, waarin alles mogelijk moet zijn. Het pakket kan tekeningen in alle formaten inlezen. Het tekenprogramma werkt zowel met pixel- als met objektgeoriënteerde tekeningen. De fonts in het TTP-pakket werken pixel georiënteerd. Het TTP-pakket wordt standaard met 14 verschillende fonts geleverd en natuurlijk kunnen de fonts van That's Write zonder problemen ingeladen worden. Verder is het mogelijk dat alle beschikbare printerfonts gebruikt kunnen worden. Werkt iemand met een printer waarvoor geen printerdriver beschikbaar is, dan wordt daar geen hand voor omgedraaid. Qua prijs zal dit nieuwe pakket natuurlijk boven de prijs van That's Write komen te liggen, maar een vaste prijs staat nog niet vast. Voor informatie: Compo 04458-1797

(js)

## Graph-IT

**Tekenen vanuit Wordplus**
Van de maker van Gracia-ProPrint hebben wij een beta-versie van het programma Graph-IT ontvangen. Dit programma maakt het mogelijk om tekeningen en grafieken te vervaardigen tijdens de tekstverwerking met (bijvoorbeeld) Wordplus. Dit wordt gerealiseerd door het programma als desk-accessory uit te voeren. De opslag van de geproduceerde tekeningen wordt via het bekende Snapshot-accessory (van Wordplus) verzorgd. Dit accessory wordt niet meegeleverd.
Het programma biedt vooral veel mogelijkheden voor het tekenen van grafieken en staafdiagrammen. Diverse lineaire en logaritmische schaalverdelingen zijn vast ingebouwd. Gewone (pixel-georienteerde) tekeningen zijn ook goed mogelijk. De bediening van het programma vergt enige gewenning, maar dat geldt eigenlijk voor elk tekenprogramma. Graph-IT ziet er op het eerste gezicht beslist aantrekkelijk uit. Het gebruik vanuit Wordplus levert weinig problemen op, maar het gebruik vanuit andere programma's soms wel. Een uitgebreide bespreking zal worden gepubliceerd als de definitieve versie van Graph-IT gereed is.

*Producent: Gracia ProSoft*
*Prijs: f 69,-*

(hd)

# De Hannover Messe - CeBit 1989

**Hier volgt een verslag van het bezoek dat twee mensen van ST hebben gebracht aan de jaarlijkse computerbeurs in Hannover. Dit verslag beperkt zich noodzakerlijkerwijs tot het ATARI ST-gedeelte van deze beurs.**

## Persconferentie

Aan het einde van de eerste dag van de Hannover-Messe werd traditiegetrouw door ATARI-Duitsland een persconferentie belegd in een hotel. De naar schatting tweehonderd genodigden, bestaande uit ATARI-dealers uit verschillende landen, computerjournalisten e.d., werden op de hoogte gesteld van de nieuwste producten uit de ATARI-stal of producten die in de nabije toekomst uitgebracht zullen worden.

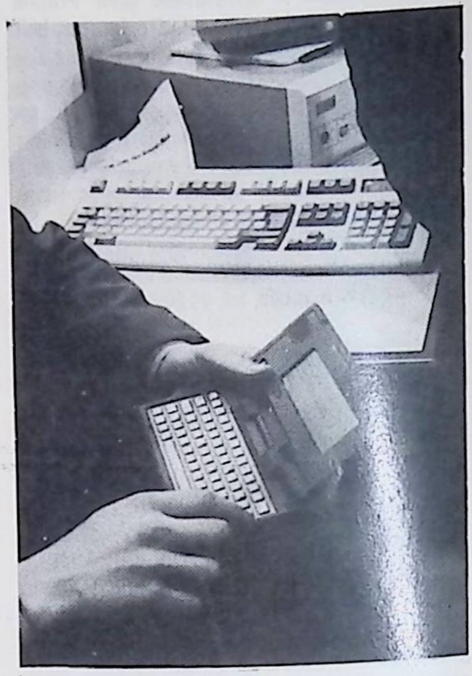
De spits werd afgebeten door de directeur van ATARI-Duitsland en tegelijk van de gehele Europese tak, Alwin Stumpf. Hij gaf de omzetcijfers van het afgelopen jaar en vergeleek deze met die van de andere jaren. Na een inzinking in 1987 gaat het weer bergopwaarts. Een van de redenen voor de problemen was het gebrek aan RAM-chips, waardoor levering stagneerde. Hij vertelde dat ATARI nu met Duitse firma Siemens een contract heeft lopen voor de levering van een zeer groot aantal geheugenchips. De continuïteit is hierdoor beter gewaarborgd en de prijs van producten kan minder snel ongunstig worden beïnvloed. De Amerikaanse ATARI-verkoop is heel duidelijk achtergebleven in vergelijking met de Europese. Kortom, in Amerika is de ST-serie nog steeds niet definitief doorgebroken, hoewel momenteel de ST's wat terrein terugwinnen als muziekcomputers; dit zou ten koste gaan van de Macintosh computer. Dat de mooiste hardware niets waard is zonder software van een gelijk niveau begint ATARI nu ook beter te beseffen. ATARI wil nu zelf een beperkte aantal pakketten van uitstekende kwaliteit gaan ondersteunen. Met name desktop publishing m.b.v. de ST-serie willen ze op de voorgrond brengen. Om het niet bij een loze kreet te laten, is ATARI-Duitsland begonnen met de opleiding van een kwart van de officiële dealers tot DTP-specialisten. Alleen maar apparaten over de toonbank schuiven is dus voor deze groep niet meer bij.

De aankondiging van de nieuwe producten was voorbehouden aan Sam Tramiel. Deze verontschuldigde zich eerst voor de afwezigheid van zijn vader, opperbaas Jack Tramiel; deze had een afspraak in Australië laten samenvallen met de Hannover-Messe en had tenslotte gekozen voor het zuiderlijk halfrond. Een andere afwezige was Shiraz Shivji, de ontwerper van de ST. Hij heeft om gezondheidsredenen enkele weken geleden voor onbepaalde tijd vrijaf genomen, vertelde Sam Tramiel de aanwezigen merkbaar geëmotioneerd. De rest van zijn praatje besteedde hij aan een uiteenzetting over het ATARI-beleid t.a.v. de microcomputermarkt; daarnaast gaf hij een algemene beschrijving van de nieuw ontwikkelde apparatuur. Enkele apparaten waren in de zaal opgesteld om bewonderd en betast te worden. We komen hier dadelijk op terug.

De technische aspecten van de nieuwe ontwikkelingen werden belicht door Leonard Tramiel, die vooral belast is met het operating system van de ATARI-computers. Helaas ontbraken in zijn betoog gegevens over de verwachte TT; dat was het nieuws waarop veel mensen eigenlijk zaten te wachten. Wel heeft ATARI te kennen gegeven het UNIX-systeem te zullen inzetten op machines die ze in de nabije toekomst hopen uit te brengen. Voor degenen die tussen de regels door willen lezen, is de conclusie wel te trekken.

## ST-achtige producten

De show werd waarschijnlijk gestolen door de Folio, een volwaardige computer van zo'n 450 gram met de afmetingen 18x9x2,7. De krachtbron vormen twee kleine batterijen die 6 tot 8 weken lang voldoende stroom kunnen leveren. Het is een PC-type computer die loopt onder MS-DOS. Hij beschikt over 63 toetsen van een zeer laag profiel. Het LC-schermpje kan 8 regels van 40 tekens bevatten; ook is het gebruikelijke 25x80-formaat beschikbaar, maar dat zal veel scrollen met zich meebrengen. De 256 Kb ROM bevat niet alleen het systeem maar ook enkele veel gebruikte typen programma: een tekstverwerker, een spreadsheet, een bestandsprogramma voor adressen en telefoonnummers, een programma dat afspraken opslaat (met wekker functie); verder zijn nog communicatiefaciliteiten en een calculator ingebouwd. Het apparaatje beschikt over 128 Kb RAM, extern uit te breiden tot 640 Kb. Uitwisseling van gegevens



*De Folio: klein en handig*

met andere computers verloopt via de gecombineerde seriële en parallelpoort. Credit cardachtige kaartjes zorgen voor externe geheugenopslag van programma's en data. Het apparaat kost in Duitsland 800 DM. De heer Stumpf sprak de veronderstelling uit dat hij die avond aan de meesten van ons deze Folio had kunnen kwijt raken als ze daar hadden gestaan.

## STACY: de draagbare ST

Onder de naam STACY heeft Atari op de CeBit haar draagbare uitvoering van de ST gepresenteerd. Op de persconferentie werd nog uitgelegd dat die naam oorspronkelijk alleen voor intern gebruik bedoeld was, maar dat het apparaat onder die naam zoveel bekendheid kreeg, dat men besloten heeft het daarop te houden. De eerste aanraking was een aangename verrassing: de STACY bleek uitgerust met een prettig aanvoelend toetsenbord. Bij het indrukken van een toets voel je duidelijk of de toets echt 'ingedrukt' is. Er is namelijk een weerstand voelbaar waar doorheengedrukt moet worden. De keuze van een LCD-scherm is gevallen op een uitvoering zonder back-light, om bij gebruik in het veld het stroomverbruik zoveel mogelijk binnen de perken te houden. Op de beurs deed het scherm prettig leesbaar aan. Dit is natuurlijk beter te beoordelen in vergelijking met andere schermen onder verschillende omstandigheden. Er kan een muis aangesloten worden, maar bij



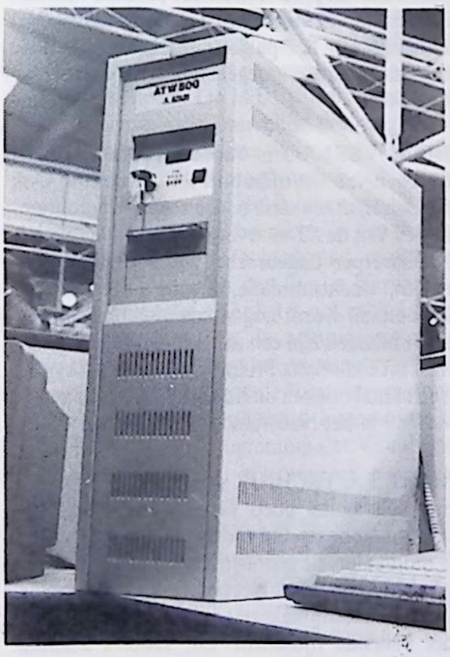
*Mega ST met CD-ROM*

gebruik op schoot is dat wat moeilijk; derhalve is het apparaat uitgevoerd met een trackball. Als het scherm dichtgeklapt is, wordt dit gefixeerd door een klepje met een veer. Tijdens het gebruik kan onder het klepje een vel papier geklemd worden. De eerste uitvoering, die kort na het verschijnen van dit blad leverbaar zou zijn, bevat twee 720 Kb diskdrives en één megabyte geheugen voor een prijs van ongeveer 3500 gulden. Later zouden er ook uitvoeringen komen met een 20 Mb harddisk voor ongeveer 4500 gulden en een model met een harddisk en 2 Mb geheugen voor ongeveer 5500 gulden.

De 1040 ST wordt vervangen door de Mega ST 1. Deze computer heeft het uiterlijk van de MEGA ST 2 en MEGA ST 4. De MEGA ST 1 beschikt over dezelfde hoeveelheid geheugen als de 1040 ST maar heeft evenals zijn grotere broers een los toetsenbord, dat volgens velen ook nog van een betere kwaliteit is dan dat van de 1040 ST. Hiermee is dan wel het meest opmerkelijke van deze MEGA ST 1 gezegd want, wie denkt dit model te kunnen gebruiken als opstapje naar de grotere (duurdere) modellen, komt bedrogen uit. Tijdens de persconferentie zei Tramiel desgevraagd dat de constructie van het mother-board een simpele uitbreiding niet toelaat. Een gemiste kans, lijkt het wel.

De oude 520 STM is nu vervangen door de 520 STFM. Dit type heeft nu het voordeel van een ingebouwde 3.5-inch drive; het RAM-geheugen is nog steeds 512 Kb.

*De nieuwe behuizing van de ATARI Transputer Machine*

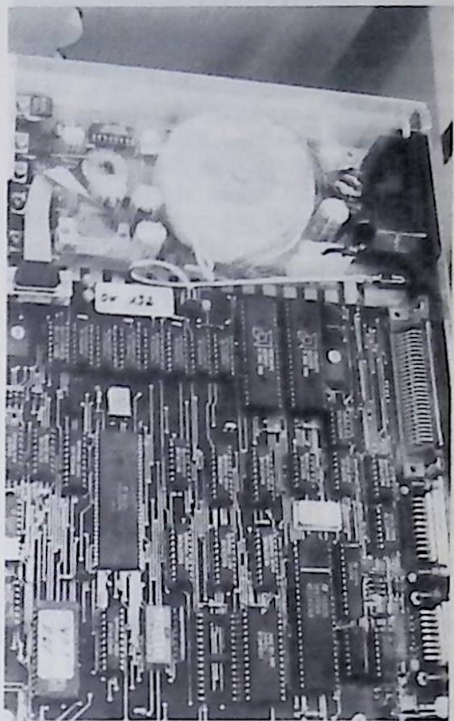


## Verwisselbare harddisk

Een nieuwe technologie die Atari gaat voeren voor zowel de ST- als de PC-productlijn is een verwisselbare harddisk met een capaciteit van 44 Mb! Door Leonard Tramiel werd gesproken van een robuust betrouwbaar apparaat, en tegen eerdere verwachtingen in is er sprake van een behoorlijk snel apparaat. De accestijd bedraagt 25 Msec en het tijdschrift Datawelt kwam bij een test van een in Duitsland ontwikkelde AT (NIET de MITAC in PCM is getest!), de PC4, tot een diskindex van 4.0 bij de Norton disktest, hetgeen in vergelijking met andere AT-disks niet slecht is. Het apparaat moet ongeveer 2500 gulden gaan kosten, de losse disks ongeveer 300 gulden. Voor degenen die behoefte hebben aan grote externe opslagruimte is er verder uitkomst in de vorm van de Megafile 30 en vooral de Megafile 60, met een capaciteit van resp. 30 en 60 Mb. Ook de verwachte CD-ROM-drive is binnenkort leverbaar. Hiermee kunnen files van enkele honderden Mb gelezen worden.

Voor degenen die nu al denken een supercomputer nodig te hebben, is er de ATW, de 'ATARI transputer workstation'. Deze computer is bedoeld om parallelle processen te verwerken. De transputer kan hiertoe worden uitgebreid met 16 extra chips, die evenzovele computers toevoegen aan het basismodel. Omdat deze computers parallel geschakeld zijn, man men hun afzonderlijke vermogens optellen om de totale capaciteit van het systeem te berekenen. De basisconfiguratie bereikt al een snelheid van 10 MIPS (miljoenen instructies per seconde). De kracht van een volledig systeem beschikt over een 'awesome power', zoals Leonard Tramiel in zijn inleiding opmerkte, waarbij er duidelijk opwinding en ook een zeker ontzag in zijn stem doorklonk. Het topvermogen ligt rond de 150 MIPS! Een van de krachtigste supercomputers van dit ogenblik, de CRAY-1, beschikt over zo'n 400 MIPS. Het besturingssyteem heet Helios. Het is, uiteraard, een multi-tasking en multi-user systeem. Wat men voor deze angstaanjagende kracht zou moeten betalen is niet duidelijk, maar wel heel wat minder dan voor de CRAY. Bij de persconferentie vroeg een journalist voor welke toepassingen dit moois nu eigenlijk bedoeld is, waarschijnlijk met de gedachte: wat heeft de gemiddelde burger aan zoveel kracht? Volgens ATARI moet men vooral denken aan real-time toepassingen en beeldverwerking. Je kunt je natuurlijk ook uitstekend een logische programmeeromgeving (b.v. Prolog) voorstellen, waarbij een aantal mogelijke oplossingspaden parallel worden afgwerkt.

Een van de producten waar de meeste aanwezigen in Hannover natuurlijk halsreikend naar hebben uitgezien, is de nieuwe 32-bit machine van ATARI, de TT (Thirty two biT). Hij was er niet. Op de vraag op welke termijn we deze nieuwe computer mochten bewonderen, werd geantwoord dat we in augustus tijdens de beurs in Düsseldorf er ongetwijfeld meer over zouden horen. Hoop doet leven. Ook de vraag of het hier ging om een UNIX-machine werd met een geheimzinnig 'could be' afgedaan. De prijs van deze machine is ook nog een groot vraagteken. Men hoopt op een basismodel (b.v. zonder UNIX) dat qua prijs niet veel duurder is dan de MEGA ST 4 op dit moment. Een aanzienlijke groep 520/1040 ST bezitters die wil uitbreiden, zou waarschijnlijk liever wachten op een TT, dan eerst nog over te gaan op een MEGA ST. Ook toekomstige kopers van een ATARI zouden wellicht een niet al te dure TT preferen boven een MEGA ST. De voorzichtigheid van de kant van ATARI is begrijpelijk, omdat eventuele (te) optimistische uitspraken over de verschijningsdatum van de TT de verkoop van de MEGA modellen behoorlijk zou kunnen verstoren.



*Hardware van snel netwerk*

## PC-achtige producten

Het is duidelijk dat ATARI niet al zijn geld op één paard wil zetten; naast de ST-serie met de MOTOROLA 680XX chips wordt ook een complete serie van vijf PC-achtige computers uitgebracht die gebruik maken van de INTEL 80XX(X) chips. Om de grafische reputatie van de ATARI computers te handhaven, worden aan de PC-typen de bekende grafische modes toegevoegd zoals de CGA, Hercules, EGA en bij de duurste modellen, de PC 4 en PC 5, ook VGA.

In de rest van dit artikel zullen we een overzicht geven van wat er zoal gedemonstreerd werd voor de ST-computers door de firma's die op de ATARI-afdeling aanwezig waren. Dit beeld is niet volledig; zelfs hier moesten we een selectie maken. De hele ATARI-afdeling is trouwens maar een fractie van wat er te zien was. Er rijden niet voor niets bussen over het terrein tussen de vele hallen.

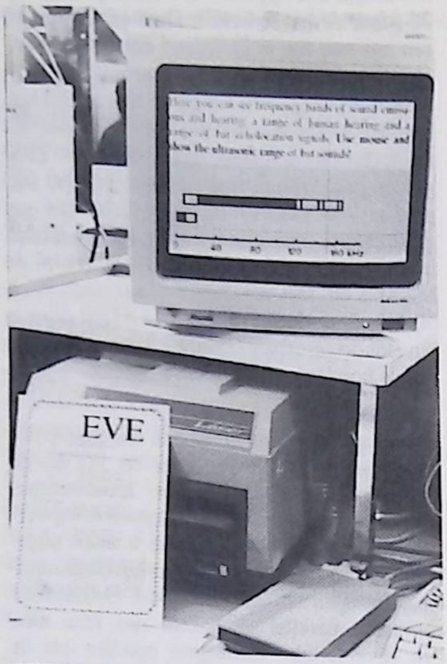
CCD is vooral bekend door programma's als TEMPUS en ST PASCAL PLUS. Zij demon-

streerden op de beurs enkele andere producten uit hun fonds zoals het pakket FTL Modula-2 en de Assembler Tutorial, een simulatieprogramma voor het leren van assembler en machinetaal. Op aanschouwelijke wijze kan men vertrouwd raken met de werking van het laagste softwareniveau van de ST: inhouden van registers en geheugenlocaties worden op het scherm getoond. Voor 98 DM krijgt men 800 pag. cursusmateriaal waaraan het programma a.h.w. is toegevoegd ter illustratie. Verder pronkte men nogal met het grafische programma MEGASTAR, dat vooral voor kunstenaars bestemd is.

Een speciale stand was ingericht om OMIKRON BASIC ten toon te spreiden. Zoals bekend is dit nu de officieel door ATARI ondersteunde BASIC. De zaak wordt fors aangepakt: er zijn al diverse grote programmabibliotheken verkrijgbaar voor o.a. statistiek, financiële berekeningen, grafisch werk, GEM-programmering en voor de MIDI-interface. Een compiler is nu ook beschikbaar. Volgens velen komt ATARI wat laat met deze ontwikkeling, omdat de BASIC-markt al heel lang door GFA-BASIC wordt gedomineerd. Voor OMIKRON pleit natuurlijk dat het compatibel is met GW-BASIC (MICROSOFT-BASIC), dat zo ongeveer de standaard vormt op dit gebied. Verder hoorden wij in de wandelgangen van ATARI-officials dat de systemen op de toekomstige machines geen rekening zullen houden met onregelmatigheden in de programmering van b.v. GFA-BASIC; de programmeurs hiervan moeten zich maar aanpassen, vindt men. OMIKRON zou hierbij niet in de problemen komen.

DATA BECKER behoeft geen introductie vanwege de veel gebruikte computerboeken voor o.a. de ST. Uiteraard werden een aantal van hun programma's gedemonstreerd:
BECKERtext voor tekstverwerking, het spreadsheet-programma BECKERcalc, en BECKERpage voor desktop-publishing; BECKERcad, tenslotte, biedt hulp bij het ontwerpen m.b.v. de computer.

APPLICATION SYSTEMS uit Heidelberg is inmiddels bij veel Nederlandse gebruikers bekend als leverancier van het tekstopmaakprogramma SIGNUM. Hoewel SIGNUM eigen fontformaat aanhoudt, is er inmiddels een reusachtige verzameling fonts ontworpen door allerlei taalspecialisten. Het is dan ook niet verwonderlijk dat er rond dit programma weer enkele hulpprogramma's ontstaan.
SCARABUS is b.v. een nieuwe fonteditor, die naast de bijgeleverde editors gebruikt kan worden; men kan sneller afzonderlijke letters ontwerpen maar ook een font in zijn geheel veranderen. Met PROTOS kunt u b.v. een deel van een SIGNUM 2 afbeelding uitvergroten in drie stappen; toets- muismacro's veraangenamen het computerwerk; er kan een grootbeeldscherm mee gesimuleerd worden en ook kunnen complexe letters gevormd worden (b.v. met een Umlaut). Omdat SIGNUM uit verschillende delen bestaat en zaken als fonts in diverse folders verstopt kunnen zitten, biedt SIGNUM!-MANAGER uitkomst: het vormt de spil van waaruit alle SIGNUM-onderdelen kunnen worden opgeroepen; maar ook kunnen enkele gewone diskettehandelingen worden uitgevoerd en daarnaast nog enkele programma's naar eigen keuze. FLEXDISK is een resetbestendige RAM-disk die flexibel is wat betreft zijn omvang. Een andere bijzonderheid is dat deze RAM-disk bootable is, m.a.w. een autofolderhet en ook accessories behoren tot de mogelijkheden.
Wat de programmeertalen betreft, profileert deze firma zich m.n. door hun MEGAMAX MODULA-2 en hun MEGAMAX C pakketten.




*STEVE op een MATRIX grootbeeldscherm*

## STEVE

De Duitse firma Kieckbusch, de officiële Duitse distributeur van STEVE, was niet aanwezig. De ontwerper en programmeur van dit zeer omvattende programma, de Joegoslaaf Primoz Jakopin, was echter zelf ter plekke om zijn product te promoten. Hij is met het ontwerp van dit programma jaren geleden begonnen op een 8 bits machine en heeft de ontwikkeling daarna voortgezet op de ST. Behalve de ene dag in de week dat hij voor de Sloveense Academie van Wetenschappen werkt, besteedt hij al zijn tijd en energie aan zijn STEVE. Hij heeft het helemaal in assembly language geschreven, iets dat bij projekten van deze omvang steeds zeldener wordt. Hij overweegt nu om te starten met het omzetten van de code naar C. Het resultaat zal dan waarschijnlijk moeten lopen op een Mega ST, gezien het beslag dat het nu al legt op het geheugen. Hij houdt niet op om er onderdelen aan toe te voegen. Zijn laatste uitbreiding betreft een leermodule, waarmee m.b.v. een klein aantal commando's lessen kunnen worden samengesteld met vragen en antwoorden. Ook de grafische mogelijkheden kunnen worden benut, waardoor aantrekkelijke combinaties van illustratie en tekst verkregen kunnen worden. We werden direct aan een proefles onderworpen over meteorologie, die we, dankzij de regelmatige uitleg door mensen als Erwin Krol, er niet slecht afbrachten. In een persbericht in dit nummer kunt u nog andere nieuwtjes over STEVE lezen.

De firma MARVIN uit Zwitserland is vooral bekend door hun HAWK-scanners. Dat is op zich niet zo bijzonder meer. Daarnaast hebben zijn een programma ontwikkeld voor optische schriftherkenning, 'optical character recognition' (OCR). Hierbij wordt een tekst als plaatje ingelezen; het programma zorgt ervoor dat de lettertekens die op deze afbeelding voorkomen herkend worden volgens de specificatie van de gebruiker. Deze laatste geeft aan wat als een apart teken moet gelden en ook wat de ASCII-waarde ervan moet zijn. De relatie tussen gescande tekens en wat de gebruiker ervan wil maken is niet 1-op-1. Voor het weergeven van letters in sommige talen kan het nodig zijn om soms drie Latijnse letters (uit de ASCII-set) te kiezen. Het programma heet in dit geval AUGUR. Het wordt geleverd in de vorm van een miniscule insteekmodule voor de ST. De luxe uitvoering kost 3000 DM, die met iets minder bedieningsgemak 2000 DM. Daar komt de scanner nog bij. Maar voor dit geld hebt u dan ook een programma dat prestaties levert. Bij wijze van test heb ik een kolom uit ons eigen blad laten inlezen. Na het scannen volgt eerst de leerfase, omdat ieder document typografisch in feite uniek is. Dit nam slechts enkele minuten in beslag. Hierna werd de ingelezen kolom naar ASCII vertaald en op schijf gezet. Voor zover ik heb kunnen zien ging dit foutloos. Ik ben nogal onder de indruk van dit programma en hoop het te mogen lenen voor een testversie in een volgend nummer. Voor mensen die veel geschreven informatie ter beschikking willen hebben in electronisch leesbare vorm is dit soort ontwikkelingen goed nieuws.

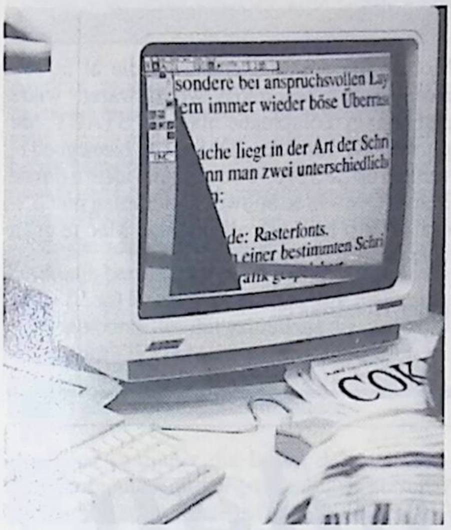
EPSILON uit Berlijn was in een afgelegen hal aanwezig. Zij zijn de huidige producenten van het bekende MPROLOG pakket, dat al enkele malen ter discussie is geweest in het ST-blad. Het wachten was op de MPROLOG compiler. Deze was al een jaar geleden aangekondigd, maar om 'technische redenen' steeds opgeschoven. Waarschijnlijk heeft men harder gewerkt aan de implementatie ervan op andere machines, omdat de belangstelling voor MPROLOG daarbij groter is. Het (complete) pakket is voor ST begrippen vrij prijzig, hoewel men moet bedenken dat precies hetzelfde product voor een mainframe tientallen malen duurder is. Het sluitstuk van het pakket zou in april aan de redactie worden toegezonden. We zijn benieuwd.

HEIM VERLAG is vooral bekend door het uitgeven van het tijdschrift ST Computer. Ook hebben zij al een serie boeken over het programmeren van de ST en over verschillende speciale onderwerpen uitgebracht (wiskundig programmeren, vluchtsimulatie, chemie e.a.). Dit boekenbestand wordt nog steeds uitgebreid. Daarnaast brengen zijn een aantal bekende programma's uit zoals Salix Prolog, ST-Math en Skyplot. Een aantal boeken uit hun fonds zal in de zomer van '89 in het Nederlands verschijnen.

MARKT & TECHNIK was duidelijk aanwezig met een aantal goed bereikbare uitstalkasten voor hun boeken- en tijdschriftenfonds. M & T is momenteel de grootste Duitse producent van computerliteratuur. Het bedrijf heeft zich ook al op het internationale vlak begeven; zo wordt b.v. het bekende tijdschrift 'Dr. Dobb's Journal'

door hun Amerikaanse vestiging uitgebracht. Voor de ST computers wordt het populaire blad 'ST Magazin' uitgegeven en er is al een hele reeks boeken uitgebracht, die de belangrijkste aspecten van de programmering van deze computers behandelen. De boeken verschijnen in gebonden uitgaven, bevatten in een veel gevallen een diskette met listings, en zijn redelijk geprijsd. Het lijkt niet waarschijnlijk dat er nog veel titels bij zullen komen, gezien de brede spreiding van hun fonds op het ogenblik.

Van hetgeen door andere firma's op de Atari-stand getoond werd overheerste het aanbod voor de professionele gebruiker en de gevorderde hobbyist. Zo waren er bij diverse stands grote (en dure) beeldschermen te zien. Wel drie firma's leverde snelle netwerken met mogelijkheden om (laser)printers en harddisks te delen met andere gebruikers. Ook pakketten als Adimens en Calamus beschikten over mogelijkheden om in netwerkomgeving te werken.

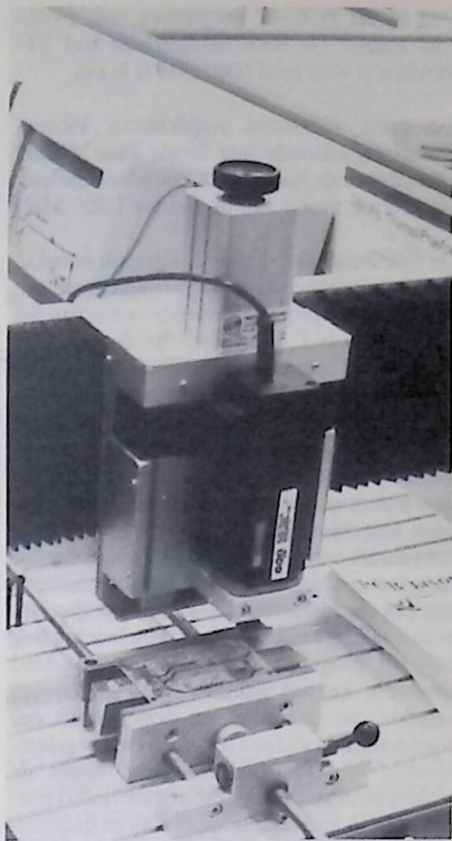



*Grootbeeldscherm aan Calamus*

De firma Computerware brengt verschillende Beckemeyerproducten die sterk beïnvloed zijn door de Unix-omgeving. Er zijn twee shell's: de MT C-Shell en een kleiner broertje, de Micro C-Shell. Nu is een shell meestal een laag om het kale operating systeem die niet veel meer doet dan het doorgeven van commando's aan het operatingsysteem. De MT C-Shell voegt echter een 'dimensie' toe, namelijk die van multi-tasking en multi-user. Een tweede gebruiker kan via de RS-232 seriële aansluiting ook op dezelfde ST werken. Met het &-teken aan het eind van een commando is het mogelijk een taak op te starten die in de achtergrond uitgevoerd wordt. Het uitprinten van files is hiervan een bekend voorbeeld, maar het is ook mogelijk om datacommunicatie op de achtergrond uit te voeren. De combinatie van een aantal producten van deze firma zou voor de fanate en verwende GEM-gebruiker een uitkomst kunnen betekenen. Er is b.v. een programma dat Visual Shell Manager heet en dat het mogelijk maakt om vanuit een GEM-omgeving, b.v. First Word, via een accessoire een window te openen waarin commando's kunnen worden gegeven aan de MT C-Shell. Er kan dan een programma worden gestart dat op de achtergrond wordt uitgevoerd, terwijl er in First Word verder gewerkt kan worden. Een ander programma in deze serie is Ansiterm, een multi-tasking VT100/VT102 terminal-emulator. Door allerlei commando's in een scriptfile te zetten kan het programma regelmatig terugkerende datacommunicatie automatisch uitvoeren, terwijl de gebruiker eventueel met iets anders bezig is.
NEODESK is meer dan alleen maar een nieuwe fileselector-box. Het biedt een desktop waarbij het aantal afgebeelde icons zich aanpast bij de ingestelde vensterbreedte. Voor wie zo langzamerhand (weer) de voorkeur geeft aan toetsenbesturing is deze utility misschien interessant. Veelgebruikte programma's kunnen in de vorm van een icon op de desktop geplaatst worden en vandaaruit gestart.
Het programma MULTIDESK maakt het mogelijk om ook na het opstarten van de computer te beschikken over deskaccessories. Men kan achteraf nog een keuze uit 256 mogelijkheden maken.

Voor professionele educatieve toepassingen brengt Weide Elektronik, bekend van de geheugenuitbreidingen voor de ST naar 1,2 of 4 Megabyte, een interface uit voor de ST aan een computerprojectiepaneel van Sharp. Dit is een doorzichtig LCD-'display' dat bovenop een overheadprojector komt te liggen. Dit apparaat kan standaard aangesloten worden op een PC/XT/AT of PS2 en kan 640*480 puntjes aansturen in 8 grijstinten. Het apparaat ziet er robuust uit en beschikt ook nog over een afstandbediening voor een robuuste prijs van 3798 DM. De eerder genoemde geheugenuitbreidingen zijn door de hoge prijs van geheugenchips nu niet erg aantrekkelijk (een uitbreidingsinbouwkaart voor de 520 en 1040 ST's kost met 2 Mb gevuld 1298 DM.) Met toekomstige lagere geheugenprijzen kunnen deze uitbreidingen weer interessanter worden.

De ST wordt in grote instellingen en bedrijven ook vaak gebruikt als low budget terminal aan grotere computers. Er bestaat echter steeds meer de behoefte aan grafische gestandaardiseerde terminals. Op het MIT Instituut in de VS speelde dit probleem sterk. Men had veel verschillende computers via allerlei netwerken aan elkaar hangen, met de behoefte deze computers onafhankelijk van hard- en software grafisch met elkaar te laten communiceren. Vanuit deze behoefte is het X Window Systeem aldaar ontwikkeld. De netwerkcommunicatie speelt een belangrijke rol bij X (zoals het systeem vaak aangeduid wordt). Het is heel goed mogelijk om drie geopende windows te hebben, waarbij window 1 voor communicatie met een lokaal programma gebruikt wordt, window 2 communiceert met een lokale mainframe en window 3 in contact staat met een supercomputer in een ver land. De firma Michael Gehret brengt onder de naam X/ST/window een X implementatie van versie 11, release 3 uit. Het is mogelijk om b.v. serieel op 19200 Baud met een Unix-systeem verbonden te zijn, waarbij een veilig en efficiënt protocol gebruikt zou zijn. Een standaardprogramma voor netwerkcommunicatie, TCP/IP, draait op de Unix-computer. De seriële verbinding kan ook vervangen worden door een Ethernet sub-systeem, maar het was op dit moment nog niet mogelijk om de ST zelf als een knoop in een Ethernet netwerk te laten fungeren.

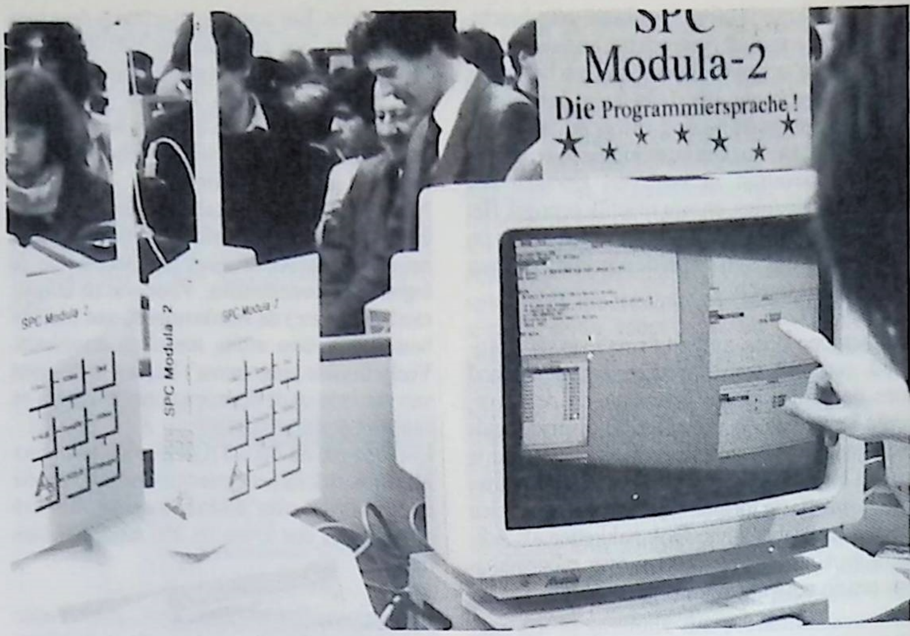


*Opstelling van PCB-layout met boorbesturing*

Op de beurs had de hard en/of software spijtig nog wat plankenkoorts
Bij het ontwerpen van complexe zaken als computers en electronische schakelingen kom je vaak het probleem van 'de kip of het ei' tegen. In veel gevallen wordt een werkende computer gebruikt om weer verbeterde opvolgers te verwekken. Zo kunnen bijvoorbeeld door simulatie van chips allerlei fouten ontdekt worden en voorspellingen gedaan worden over de werking van de chip al voor hij het eerste levenslicht heeft gezien. Zolang de chips nog niet zover zijn geïntegreerd dat je er met maar 1 kunt volstaan, heeft een computer een printplaat nodig om alle chips op de juiste manier te verbinden. Vroeger werden printplaatontwerpen met de hand geplakt. Tegenwoordig gebeurd dit ontwerpen veel op de computer, die met veel zaken kan helpen. Voor de ST waren al ruime tijd een aantal van dit soort ontwerpprogramma's. Op de beurs was een opstelling van de firma Thomas Praefcke Computer hard- en software aanwezig, waarmee niet alleen de printlayout ontworpen kon worden, maar ook kleine aantallen dubbelzijdige printen geboord en gegraveerd kunnen worden. Met het PCB-layout-programma kunnen printen van maximaal 320 x200 mm ontworpen worden met een oplossend vemogen van 1/20 inch. Naast de versie die op de beurs getoond werd, komt er medio juni een plus versie, die over een auto-router beschikt en verder de mogelijk heeft om een onderdelen-layout en een stuklijst te genereren. De standaard versie heeft alleen een route mogelijkheid voor iedere keer één verbinding. Het programma kan een boorlijst generen die weer ingelezen kan worden door het PCB-NC programma, dat de computergestuurde boor/freesmachine aan kan sturen. Het standaardprogramma kost 199,50 DM, de plus versie 348,-

DM en het PCB-NC programma gaat 1498,- DM kosten. De boor/freesmachine met ST-interface is voor rond 6000 DM te koop.

De firma Advanced Applications Viczena GmbH presenteerde hun Single Pass Modula Compiler en de Spectre 128, een Mac-emulator die gebruik maakt van de nieuwe 128K Mac-ROM's.
De SPC Modula compiler is eigenlijk ontstaan als een bijproduct in de vervaardiging van een groot softwareproject. De eerste versie van het project was is C geschreven en de gebruikte Megamax C compiler had destijds bij debuggen zoveel problemen opgeleverd, dat men besloten heeft op Modula-2 over te stappen. Hiervoor was het nodig om eerst de laatste versie van Wirth's single pass Modula compiler naar de ST over te zetten, daar men verwachtte dat deze het beste foutvrij zou zijn. Op de Atari-stand werd het gebruik van de compiler getoond in combinatie met een Matrix M110 19 inch beeldscherm met een resolutie van 1280 bij 920 puntjes, wat er zeer indrukwekkend uitzag. Een beschrijving van het programma zal elders in het blad gevonden kunnen worden. De heer Viczena vertelde dat de firma een policy heeft om regelmatig met updates te komen en zeer geïnteresseerd is in een wisselwerking tussen de makers en de gebruikers, die met allerlei ervaringen en vaak interessante applicaties en hulpmiddelen komen. Ter ondersteuning van het onderhoud van grote software-projecten werd er gewerkt aan een soort revision control system, dat code en documentatie van verschillende versie gestructureerd gaat bijhouden. Er zijn




*SPC-Modula met grootbeeldscherm*

plannen om het ontwikkelen van Turbo C programma's binnen de SPC-Modula programmeeromgeving mogelijk te maken. De editor moet ook Turbo C fouten en regelverwijzigingen gaan begrijpen, waardoor de cursor iedere keer bij de volgende fout in de tekst komt te staan. Er werd nog gewerkt aan het implementeren van virtueel geheugen onder Modula, waarna interactief debuggen mogelijk moet worden, in plaats van te moeten spitten in een post mortem dump.

Met de Spectre Mac-emulator, die al in een recente start besproken werd, waren wilde plannen in combinatie met de STACY, de portable ST van Atari. De Spectre bleek goed te werken op de STACY en er waren ideeën om de Mac-ROM's in te bouwen en tezamen met b.v. een 40 Mb harddisk als portable Mac te gaan voeren!!

**Peter Hendriks** en
**Tom den Duijf**

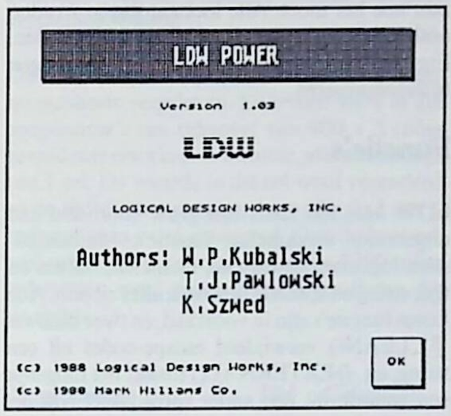
# LDW Power

## Worden mooie beloften waargemaakt?

**Op de verpakking van LDW POWER staat een aanstormende stoomtrein, waarschijnlijk bedoeld om het begrip 'power' vorm te geven. Achterop die verpakking wordt het gebruik van superlatieven niet ontweken. 'LDW POWER is far and away the most advanced spreadsheet available for the Atari ST line of computers'... 'It's fast'...'It's easy to use'...'Its versatile'...'LDW POWER is Lotus 1-2-3 Release 2 compatible and can load files to and from Lotus'.**

Al deze mooie woorden vormen een goede reden dit nieuwe spreadsheet-pakket eens extra nauwkeurig onder de loep te nemen. Persoonlijk heb ik namelijk slechte ervaringen met een ander pakket dat Lotus 1-2-3-compatible beweerde te zijn: VIP Professional. Het uitwisselen van Lotus-file naar VIP en terug heeft mij heel wat hoofdpijn bezorgd. Over de snelheid (traagheid) van VIP wil ik weinig woorden vuil maken. Ik kan me voorstellen dat gebruikers die een Atari samen met VIP kochten om thuis 'compatible' te zijn met de zaak er spijt van hebben gekregen dat ze toch geen PC aangeschaft hadden.

Voor een computer die boven de status van 'homecomputer' uit wil groeien naar 'personal computer' (zoals Atari zelf de ST betitelt) zal professionele software beschikbaar moeten zijn. Wat CAD, Midi en Desktop-Publishing betreft, is de Atari ST goed voorzien. Maar bij de 'basics' (tekstverwerking, database management en spreadsheets) ontbreekt er wat. Elk tekstverwerkingspakket op de Atari heeft wel 'iets', maar er komen weinig database-pakketten in de buurt van de snelheid die vergelijkbare programma's op de PC halen, en op het gebied van spreadsheets is er ook altijd een achterstand geweest. Kan LDW POWER dit gat vullen?




### Industrie standaard

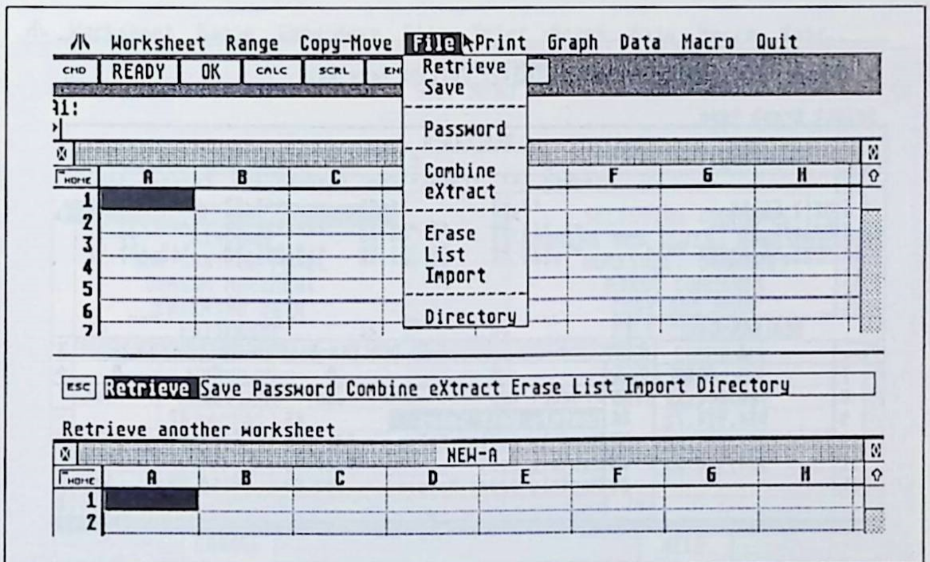
De meest belangrijke troef van LDW is compatible zijn met Lotus 1-2-3. Zoals bekend is Lotus de industriestandaard op spreadsheet-gebied. Lotus 1-2-3 Release 2 staat al maanden aan de top van de beste verkochte pakketten in de USA. Dit heeft als resultaat dat er veel boeken, kennis en ervaring over Lotus beschikbaar is. Natuurlijk is het weinig vernieuwend om een 'spreadsheet-kloon' uit te brengen, maar het helpt wel de gebruikers die er direct mee aan de slag willen. Is LDW werkelijk Lotus compatible?

### User interface

Om te kijken of de bediening van LDW hetzelfde was als die van Lotus ben ik er meteen, zonder de handleiding te lezen, mee aan de slag gegaan. Ik heb uitgebreide ervaring met Lotus, en was benieuwd of de overstap erg groot zou zijn. Bij het opstarten toont LDW een pull-down menu, GEM-stijl dus. Tot mijn verbazing ging dat menu bij het indrukken van de '/' toets over in een Lotus-stijl menu, met exact de indeling zoals ik die gewend was. Een 'lotus-menu' is ondertussen een begrip in de PC-wereld. De optie's staan horizontaal naast elkaar, en kunnen met de muis of de cursortoetsen geselecteerd worden. Ook kan de eerste letter van een gewenste optie ingedrukt worden om deze te kiezen. Na het intoetsen van '/FR' (File, Retrieve) kwam keurig een file- selection-box tevoorschijn.

### Type-ahead-buffer

LDW is een van de weinige Atari programma's die ik ken die een goed werkende type-ahead-buffer heeft. Als je bij 1st Word toetsen indrukt op het moment dat het programma bezig is, dan hoor je het 'tik'-geluid wel, maar de toetsaanslagen hebben geen effect. LDW onthoud alle toetsaanslagen, ook als het bezig is een scherm op te bouwen. Daardoor kan men met vooruit tikken een menu- selectie maken nog voordat het menu op het scherm verschenen is. Door b.v snel '/WCS8[Enter][Enter]' in te drukken verander je de breedte van de huidige kolom in 8 tekens. Na het ingeven van de laatste toets is LDW zeker nog een seconde bezig met het tekenen van menu's, maar de gebruiker kan ondertussen verder. Door deze mogelijkheid heeft de gebruiker veel minder last van de langzame GEM-interface.




*de GEM- en Lotusmenu's beide beschikbaar*

### Functietoetsen

Zelfs de indeling van de functietoetsen is hetzelfde als bij Lotus. De belangrijkste (Calc, Edit,

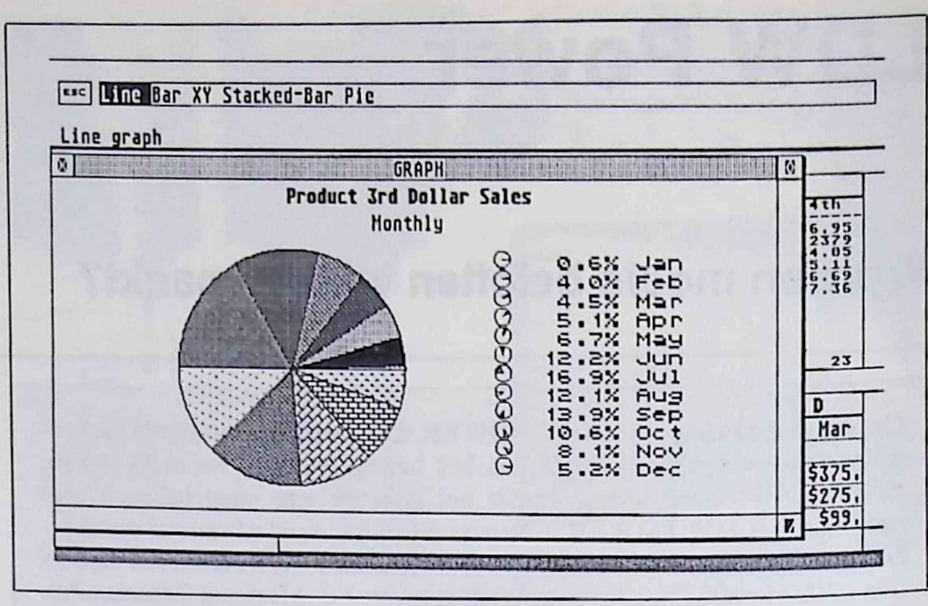
Help) zijn ook als 'button' op het scherm aanwezig, en kunnen met de muis geselecteerd worden. De enige uitzondering is F6; deze toets wijkt af. Bij Lotus is dit de [Window] toets, om van window te verwisselen dus. Op de Atari is deze toets natuurlijk overbodig: een window kan geselecteerd worden door er met de muis in te 'klikken'. Om het gat te vullen heeft LDW van F6 de [Note] toets gemaakt. Door deze in te drukken kunnen twee regels kommentaar bij de huidige cel ingegeven worden. Handig is dit vooral bij macro's, zodat men later kan lezen waar een macro voor bedoeld was.

## Lotus bestanden

De bediening was dus vrijwel gelijk, maar konden Lotus spreadsheet bestanden zonder problemen gelezen worden? De eerste poging mislukte, toen ik probeerde een Lotus versie 1a .WKS bestand te laden. LDW liep zelfs vast, zonder foutmelding. Een beetje mijn fout, want LDW beweert Lotus Release 2 compatible te zijn. Deze (nieuwe) versie van Lotus schrijft .WK1 bestanden die niet uitwisselbaar met het (oude) .WKS formaat. Lotus release 2 kan echter wel .WKS bestanden inlezen. Door dit te doen, en de spreadsheets als .WK1 bestanden weg te schrijven, kon ik aan de tweede poging beginnen. Alles ging toen vlekkeloos, niet alleen de data maar ook cel-format, column-width en andere informatie werd zonder problemen ingelezen. LDW herkende automatisch het Lotus-formaat; er waren geen conversie stappen nodig. Om een LDW spreadsheet naar Lotus terug te zetten is wel een conversiehandeling nodig. Hiertoe wordt een programma meegeleverd. Ook dit verliep zonder problemen, Lotus las de bestanden zonder te klagen. Door middel van het File Import, en het Data Parse commando kunnen ook gemakkelijk cijfers uit tekstbestanden (uit andere programma's) ingelezen worden.

## Macro's

Een sterk punt van Lotus is de macro-taal, waarin complete programma's geschreven kunnen worden. Om te testen of LDW ook macro-compatible is schreef ik in Lotus een spreadsheet-benchmark-macro. Deze maakt een 500 regels bij 11 koloms sheet aan, gevuld met onderling afhankelijke getallen (genoeg om 500k geheugen te vullen). Elke cel in de sheet bevat dus een rekenkundige formule, met een referentie naar een andere cel.
Vervolgens kopieert, verplaatst en verwijdert de macro complete kolommen. Deze macro werd gebruikt om LDW en Lotus in snelheid met elkaar te kunnen vergelijken. Het bleek zonder problemen op de ST te werken, omdat ik alleen de '/X' commando's gebruikt had. Lotus release 2 bevat ook 'macro keywords'. Deze heeft LDW niet in huis, een van de weinige punten waarbij LDW afwijkt. De '/X' commando's komen van Lotus versie 1; je zou dus kunnen zeggen dat LDW Lotus release 1 macro-compatible is. Overigens kent LDW een paar extra macro-commando's waarmee het scherm 'bevroren' kan worden tijdens de uitvoering van macro's. Dit heeft, zoals later besproken zal worden, een geweldige snelheidsverbetering tot gevolg.




LDW bevat overigens net zoals 1-2-3 een 'learn-mode'. Hiermee kunnen macro's gemaakt worden door simpelweg de computer te laten zien hoe het moet. Alle toetsaanslagen worden onthouden, en kunnen later 'afgespeeld' worden. Erg handig om steeds terugkerende handelingen te automatiseren.

## Functie's

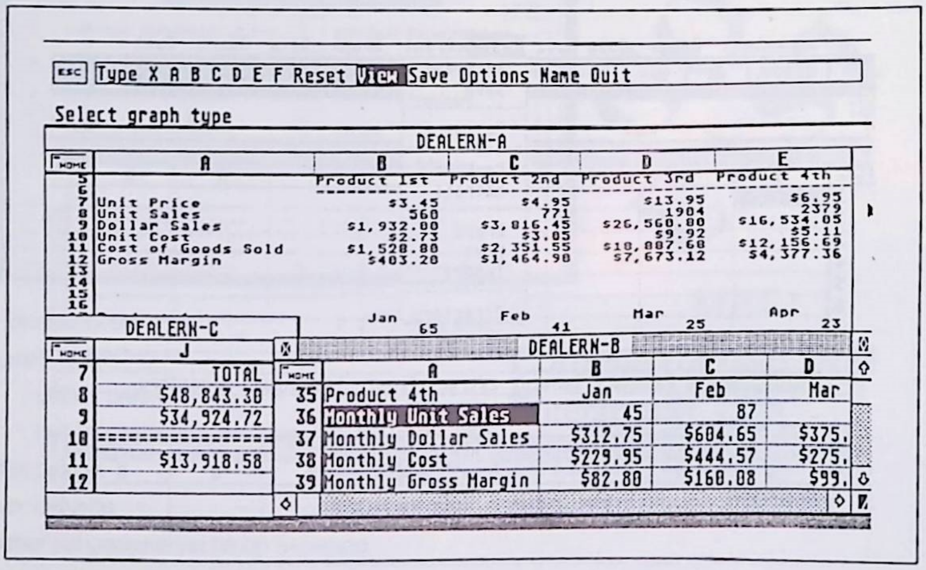
LDW kent net zoals zijn grote voorbeeld een uitgebreide verzameling functie's. Mathematische, logische, statistische, financiële, datum en tijd, string en speciale functie's: alles zit erin. Alle Lotus functie's zijn in voorraad, en twee nieuwe. @CLEAN() verwijderd escape-codes uit een string, en @EXTERNAL() maakt het mogelijk een waarde uit een ander spreadsheet (op het scherm of op schijf) in de huidige spreadsheet op te nemen. Ook de parameters die aan elke functie meegegeven moeten worden, zijn hetzelfde. Elk boek over Lotus kan dus zonder problemen bij LDW POWER gebruikt worden.

## Commando's

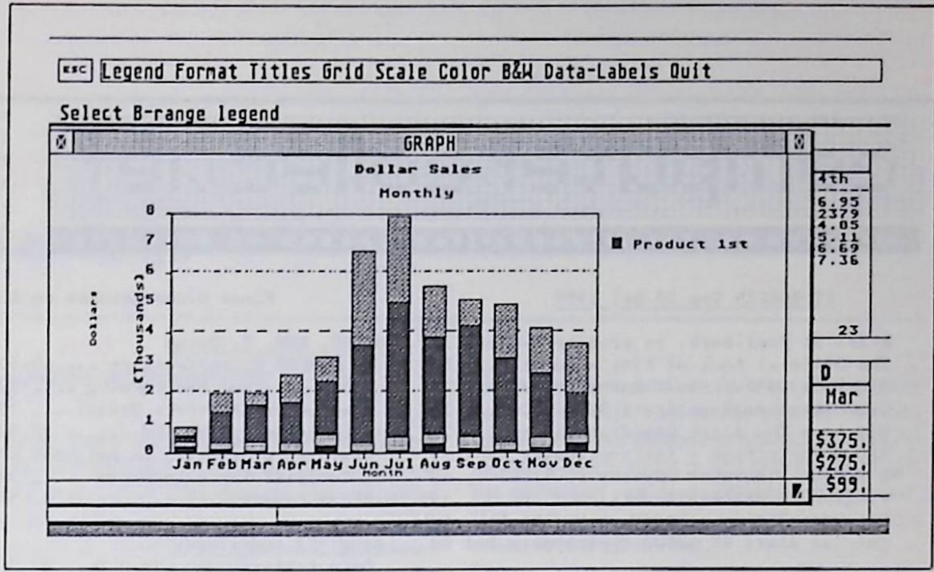
Alle commando's van Lotus (zoals File Retrieve) zitten in LDW, behalve Data Matrix, en Data Regression. Met de eerste kunnen in Lotus matrices geïnventeerd en vermenigvuldigd worden, met de tweede kan regressie-analyse uitgevoerd worden. Ook LDW heeft enkele mogelijkheden die Lotus niet heeft. Onder andere kunnen grafieken afgedrukt worden, zonder dat daar een speciaal programma voor nodig is. Grafieken kunnen ook in een meta-file opgeslagen worden. Zo'n meta-file kan door sommige andere programma's gelezen worden, zodat het mogelijk is de grafieken van de ene naar de andere applikatie over te zetten. Voordat grafieken afgedrukt worden kan een x/y schaal ingegeven worden. Hiermee kan de breedte-hoogte verhouding (aspect ratio) veranderd worden.

Een belangrijk verschil met Lotus is dat LDW onder GEM werkt. Verschillende, van elkaar afhankelijke spreadsheets kunnen tegelijkertijd op het scherm in een window geplaatst worden. Ook kan een grafiek tegelijk met de bijbehorende data op het scherm gezet worden. Elke verandering in de data wordt meteen in de grafiek zichtbaar gemaakt.

Zogenaamde 'ranges' (groepen cellen waarmee een bewerking uitgevoerd kan worden) kunnen



met de muis aangewezen worden. Door de [Control] toets ingedrukt te houden tijdens het selecteren van een range kiest men automatisch voor het Move commando. Men hoeft dan alleen nog maar aan te wijzen waar de range naar toe verplaatst moet worden. Door bij het aanwijzen de [Alternate] toets vast te houden kiest men voor het Copy commando. LDW blijkt zeer flexibel, men kan eerst voor b.v. het Range Format commando kiezen, om dan de Range aan te wijzen, of men wijst eerst de range aan om vervolgens voor Range Format te kiezen. Beide mogelijkheden leiden tot het zelfde resultaat, de gebruiker mag zelf kiezen welke aanpak het prettigste uitkomt. Al dit soort details maken het werken met LDW zeer plezierig. Alhoewel het woord door frequent misbruik veel betekenis verloren heeft, wil ik het toch wagen LDW POWER 'gebruikersvriendelijk' te noemen.




## Snelheid

Een belangrijk punt is de snelheid. LDW ziet er goed uit, maar hoe zit het met de 'ultra-fast response time' die beloofd wordt? Hiertoe is de hiervoor vermelde (en op het STER BBS opvraagbare) macro op een IBM AT (6 Mhz) en op een Atari 1040 ST gedraaid. Omdat tijdens het uitvoeren van een macro bij LDW het scherm 'bevroren' kan worden, heb ik de test een keer met en zonder deze optie gedaan. Het resultaat staat in de volgende tabel.

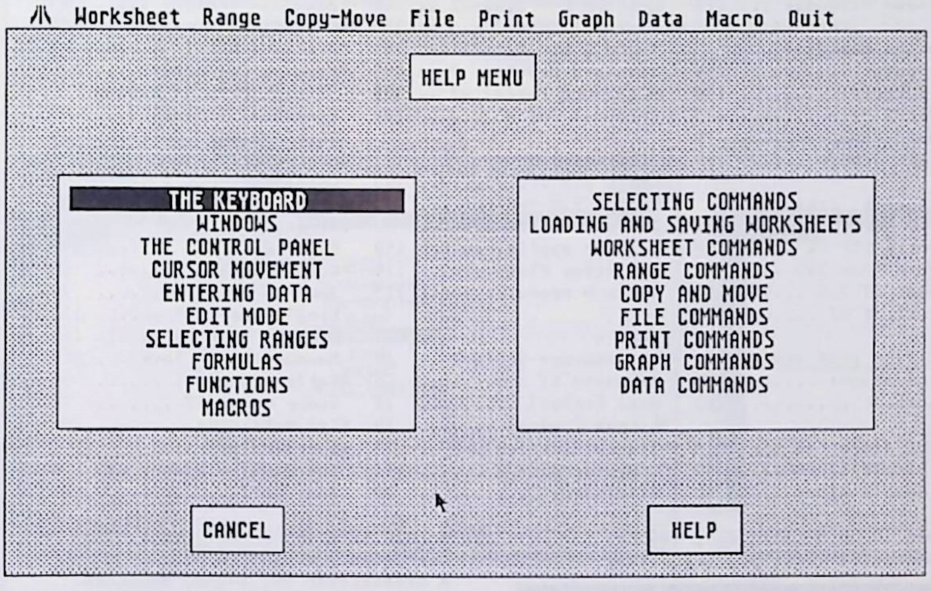
| | |
|---|---|
| Lotus 1-2-3 | 104 sec |
| LDW (output on) | 216 sec |
| LDW (output off) | 59 sec |

Het tonen van de gegevens op het scherm is dus de vertragende faktor bij LDW. Dit hoeft niet te verbazen: Lotus op de PC zet de uitvoer direct in het geheugen van de video-kaart (direct screen IO) terwijl de Atari het scherm grafisch op moet bouwen. Als de berekeningen 'achter de schermen' plaats vinden, dan wint LDW het zelfs van Lotus. Door LDW op een Mega-ST met blitter uit te proberen werd deze conclusie nog eens gestaafd. De verwerkingstijd met 'output on' liep terug tot 116 seconden, terwijl met 'output off' nog steeds 59 seconden nodig was. Het grotere geheugen van de Mega ST bracht dus geen versnelling teweeg.

Ook heb ik LDW met twee andere Atari spreadsheets vergeleken. Daarvoor werd in alle programma's een rekenvel van 900 x 3 cellen gevuld met een simpele formule, alle afhankelijk van 1 cel. De waarde in die cel werd veranderd, en de tijd die het programma nodig had om te herberekenen werd genoteerd. Ook de hoeveelheid geheugen die nodig was om de 2700 cellen te bevatten werd genoteerd. De resultaten zijn als volgt:

| Programma | Tijd | Geheugen |
|---|---|---|
| LDW POWER | 34 sec | 396k |
| VIP Professional | 48 sec | 351k |
| Logistix | 95 sec | 152k |

'Ultra-fast response time' is misschien overdreven, maar van alle geteste spreadsheets op de Atari is LDW POWER zeker de snelste. Zuinig met het beschikbare geheugen is LDW echter niet. Voor dezelfde spreadsheet had Logistix maar de helft van het geheugen nodig. Als men van plan is met zeer grote modellen te werken, maar men heeft geen geld voor een Mega-ST en men neemt een afstotende user-interface voor lief, dan kan men beter voor Logistix kiezen.




## Samenvatting

Zoals hiervoor vermeld is LDW inderdaad snel. Zowel het opbouwen van het scherm als het herberekenen van grote werkvellen gebeurt binnen een acceptabele tijd. Het programma is ook zeker gemakkelijk in het gebruik, mede dankzij het gemakkelijk toegankelijke hulpsysteem. Alle mogelijkheden zijn met de muis of via het toetsenbord te kiezen, zodat de gebruiker het programma kan bedienen zoals hij/zij gewend is. De zeer sterke gelijkenis met Lotus 1-2-3 maakt het mogelijk instructieboeken over Lotus ook bij dit programma te gebruiken. Het programma is tevens 'versatile' te noemen. Gegevens kunnen gemakkelijk met andere programma's uitgewisseld worden. Grafieken kunnen zonder veel moeite gemaakt, afgedrukt en als meta-file bewaard worden. Doordat LDW met GDOS werkt is het printen van de grafieken op een Atari laserprinter geen probleem. De database-functie maakt het mogelijk te selecteren, sorteren en grote hoeveelheden informatie te analyseren. En met de 'macro-recorder' kunnen standaard handelingen geautomatiseerd worden.

LDW POWER maakt de meeste beloften dus waar. De bij het pakket horende handleiding ziet er goed en overzichtelijk uit. Als men een spreadsheet zoekt voor de Atari ST, dan is dit er een die zeker in overweging genomen moet worden.

**Berg Chabot**

# computercollectief

**microcomputer tijdschriften boeken en software**

*******************************************
* onze nieuwe VOORJAAR '89 CATALOGUS *
* is nu uit. We sturen hem GRATIS toe *
* als je ons een kaartje stuurt met *
* je naam en adres. Vermeld tevens *
* 'STICHTING ST' *
*******************************************

## ST BOEKEN Top 30 Mei 1989

Atari ST Profibuch, 5e erweiterte Aufl 79
The Official Book of King's Quest .... 35
Hintbook Leisure Suit Larry (Sierra) . 20
Gebruikershandboek Atari ST .......... 25
Hintbook The Black Cauldron (Sierra) . 20
68000 Pocketbook - instructionset,6801 15
*A Flight Simulator Odyssey (Gulick) .. 45
Program in GFA BASIC version 2 and 3 . 39
Das grosse Signum! Buch - incl disk .. 69
Concise Atari ST 68000 Programmers Ref 69

Het grote BASIC-boek voor de Atari 59,90
*De Motorola MC 68000-familie (Harman) 85
Data Becker Führer - Signum! ......... 35
*MC68030 Enhanced 32-Bit Microprocessor 79
Flying Flight Simulator with MAC, ST . 29
*Hintbook Manhunter (Sierra) .......... 20
Learning C on the Atari ST (Wikert) .. 55
Das grosse GfA-BASIC 3.0 Buch inkl com 59
Technical Reference Guide ST - vol 3 . 55
Technical Reference Guide ST - vol 2 . 55

Das MIDI- und SOUND-Buch zum Atari ST 79
Technical Reference Guide ST - vol 1 . 55
*Atari ST C Know How : Laser-, Megamax- 49
GFA-BASIC 3.0 Training ............... 39
Atari ST Introduction to MIDI Programm 55
Software-ontwikkeling Atari ST - GEM, 69
GfA BASIC - GfA-Buch ATARI ST +disk .. 99
*Atari ST 3D-Grafik und Animation ..... 79
MIDI and Sound Book for the Atari ST . 49
Data Becker-gids: GfA BASIC 3.0 ... 37,90

## Nieuw Binnengekomen en Actuele ST Boeken

### 68000, GEM, C, Occam

De M68000 Microprocessor ... 49,50
68000 Ass Lang Programming .... 59
Transputer Reference Manual ... 79
*Data Becker-gids: GEM ...... 32,90
Learning C:Graphics on Amiga&ST 49
A Tutorial Intro to Occam ..... 39
Occam Programming ............. 58

### Atari ST nederlands

Data Becker-gids: GfA 3.0 .. 37,90
Software Ontwikkeling op de Atari ST (Geiss) ........... 69,00
Het GEM boek voor de ST .... 59,90
Machinetaalboek voor de ST . 49,90
Tips en Trucs voor de ST ... 59,90
Grote BASIC boek voor ST ... 59,90

### Atari St duits

*Atari ST C Know How ........... 49
Laser-,Megamax- und Turbo-C
*ST 3D-Grafik und Animation .... 79
incl. disk
Das Adimens Praxis-Buch ....... 69
Programmierpraxis OMIKRON BASIC 69
Data Becker Führer - Adimens .. 35
Data Becker Führer - Omikron .. 30
Data Becker Führer - Signum! .. 35
Data Becker Führer - 1ST Word . 30
Das Grosse GFA-BASIC 3.0 Buch . 59
ST Besten Tips und Tricks+dsk . 69
*Atari ST Intern - Band 2 ...... 89

### Atari ST - Duits, Engels

Das Grosse SIGNUM! Buch +disk . 69
GFA-BASIC 3 Referenz Handbuch . 69
GFA-BASIC Advanced Programming 79
Program in GFA-BASIC 2.0 & 3.0 39
Learning C on the Atari ST .... 55
Atari ST 1st Word Plus + disk . 59
Atari ST assembler Buch +disk . 75
Technical Reference ST -3 TOS . 49

FlightSimulator Adv for MAC, ST 35
Gunship Academy ............... 39
Official Book of King's Quest . 35

### Virussen

Das grosse Computer-Viren Buch 59
Compute!'s Computer Viruses ... 45

### SIERRA Hint Books

*Hintbook Black Cauldron ....... 20
*Hintbook Space Quest .......... 20
*Hintbook Police Quest ......... 20
*Hintbook Leisure Suit Larry ... 20

### MIDI boeken

Muziek m.d.Computer inc MIDI 29,[illegible]
*MIDI and SOUND Book for the ST [illegible]
Atari ST Introduction to MIDI . [illegible]
MIDI - the Ins, Outs & Thrus .. [illegible]
The MIDI Resource Book ........ [illegible]
The MIDI System Exclusive Book [illegible]
Mind over MIDI ................ 69

## A C T U E L E   S T - S O F T W A R E   (inclusief BTW)

### ST Software TOP 30

Falcon - the F-16 Fighter 95
Turbo C mit MAS-Assem ... 399
Wercs - Resource Constr.. 115
Leisure Suit Larry II ... 115
Operation Wolf (color) .. 79
STOS - Game Creator ..... 119
*Police Quest II ......... 95
GfA BASIC 3.0 interpreter 269
GfA Utility Kleurconverter 69
*F-16 Combat Pilot (color) 95

*De 20 Beste GFA Programmas 79
Gunship - helicopter .... 89
Starglider 2 (color/mono) 89
1ST Word Plus version 3.0 329
*Gold Rush ............... 95
*Populous (color) ........ 95
*Double Dragon (color) ... 79
Utilities Plus .......... 115
Scenery Disk West Europe 59
Zany Golf (color) ....... 89

TurboST -software blitter 89
*Hit Disks - volume 2 .... 95
Elite (colour) .......... 95
Dungeon Master (color) .. 89
Outrun (color) .......... 79
5 Star Compilation ...... 95
Shadow -file transfer ... 99
GfA-Assembler .......... 189
Flash 1.6 datacommunicatie 99
Music Construction Set .. 39

### ST toepassingen :

DC Boek -boekhouden .. 159
Marko Manager boekhoud 369
*Mavis Beacon Typing .. 149

communicatie:
Interlink ST ......... 119
Shadow -multitasking .. 99

databases ed:
Superbase Professional 799
Superbase Personal ... 229

grafiek en muziek:
CyberSculpt .......... 239
CyberTexture ......... 139
*GFA DRAFT Plus ....... 439
ST Sound-Enhancer .... 99
MidiSoft Studio ...... 199

spreadsheets, tekstverw.:
Graphic Sheet ........ 179
TimeWorks DTP ST ..... 449
1ST Word Plus 3.0 .... 329
Textomat ST 3.0 ...... 125
WordPerfect ST ...... 1179

mini office prof serie:
communications ....... 95
spreadsheet .......... 95

*NASHUA diskettes per 10 *
3.5" single sided ..... 39
3.5" double sided .. 42,50

### ST programmeertalen :

68000 Tutor/Simulator 125
Hisoft DEVPAC ST 2.0 . 235
Metacomco Assembler 12 219
GFA Assembler -nederl. 189
GFA BASIC 3.0 ........ 269
PowerBASIC compiler .. 199
Hisoft BASIC compiler 319

Aztec C68 Standard ... 499
Azetc C68 Developers . 679
Lattice C - 3.04-02 .. 349
Mark Williams C 3.0 . 359
Turbo C mit Assembler. 399
Modula-2 Develop. Vers 495
Metacomco Pascal v2 .. 329
Personal Pascal ..... 299

utilities :
Craft 2.0 ............ 395
Tempus 2.0 -editor ... 159
TurboST -softw blitter 89
Twist - applic. switch 159
Utilities Plus ....... 115
Wercs - resource const 115

games :
Afterburner (color) .. 79
Arkanoid II .......... 79
Baal (color) ......... 79
*Batman (color) ....... 79
*Blasteroids (color) .. 79
Computer Hits 4 ...... 115
Eliminator ........... 79

### ST games:

Double Dragon ........ 79
*F-16 Combat Pilot .... 95
FOFT ................. 115
*Hit Disk vol 2 ....... 95
Hostages (color) ..... 95
Kennedy Approach (color)95
Lombard RAC Rally .... 95
Menace ............... 79
Operation Wolf (color) 79
Pacmania ............. 79
*Populous (color) ..... 95
Powerdrome ........... 95
*Precious Metal -4 games 95
*Premier Collection ... 115
Zynaps,Nebulus,Exolon
Speedball (color) .... 95
Thunderblade (color) . 79

denkspelen:
Colossus Chess X ..... 95
DIOS Dammen (mono) ... 89

adventures:
Black Cauldron ....... 149
*Gold Rush ............ 95
Joan of Arc .......... 79
King's Quest IV ...... 115
Leisure Suit Larry II. 115
Manhunter New York ... 115
*Police Quest II ...... 95
Space Quest II ....... 79
*Zak McKracken ........ 95

sportsimulaties:
The Games - Winter Ed. 79
Zany Golf ............ 89

winkel open van dinsdag t/m zaterdag tussen 10 en 5 (maandag gesloten)
alle prijzen zijn inclusief BTW - verzendkosten f 6,- per bestelling

Amstel 312 (t.o. Carré) | 1017 AP Amsterdam | Fax (020) 226668 | Postbank 4475158 | NMB 697915646

in BELGIE is alles verkrijgbaar bij:
Het Computerwinkeltje pvba
M Sabbestraat 39, B-2800 MECHELEN
fax: 015-207 332 tel: 015-206 645

dealer aanvragen welkom

# Analyze!

## Een spreadsheet zonder franje

Een van de weinige goede argumenten om een PC te kiezen in plaats van een ST was het ontbreken van een echt goed spreadsheet-programma voor de ST. Van diverse kanten wordt nu getracht om ook dit argument zijn kracht te laten verliezen. De nieuwe generatie spreadsheets is beter en sneller dan het oude vertrouwde (?) VIP-Professional. In dit artikel wordt ANALYZE! onder de loep genomen. We moeten er echter op wijzen dat het hier gaat om een Beta-versie in die zin, dat het programma pas omstreeks september officieel op de markt wordt gebracht.

Het bekende Lotus 1-2-3 spreadsheet heeft model gestaan voor veel andere spreadsheets. Helaas is men daarbij meestal uitgegaan van de eerste versie van Lotus 1-2-3. De tweede versie heeft namelijk veel meer mogelijkheden en is ook iets consequenter van opzet. Analyze! is min of meer gebaseerd op de eerste versie, maar kan beslist geen kloon worden genoemd. Er zijn duidelijke afwijkingen die verderop in dit artikel ter sprake zullen komen.

### Uitvoering

Analyze! (2 diskettes) wordt samen met de handleiding geleverd in een plastic zak. De eerste diskette bevat het programma zelf plus enkele zeer eenvoudige demonstratiefiles. De tweede diskette bevat drivers voor de laserprinter. De handleiding (in de Engelse taal) is duidelijk en overzichtelijk. Ook beginners worden goed op weg geholpen. De 'help'-faciliteiten van het programma zelf laten de gebruiker volledig in de kou staan. In feite wordt na het indrukken van de Help-toets vrijwel alleen een verwijzing naar de menubalk op het scherm getoond.

### Layout en bediening

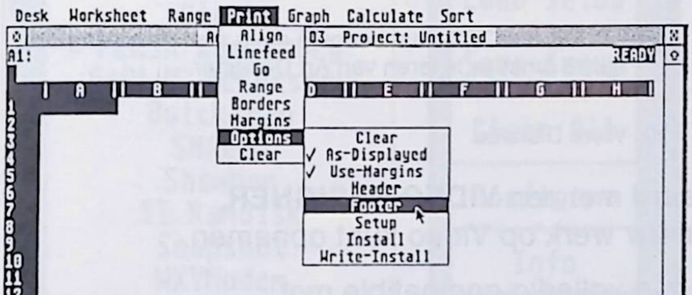
Na het starten van het programma verschijnt een scherm dat mij enigszins aan de eerste versie van VIP-Professional doet denken. Het rekenblad wordt echter wel omgeven door een GEM-venster. Er worden 17 rijen en 8 kolommen getoond. Op een monochroom scherm laat het venster zich zodanig vergroten dat er 18 rijen worden getoond. Het merkwaardige is dat dit alleen lukt met de 'size box' (rechtsonder) en niet met de 'full box' (rechtsboven). Meer dan 18 rijen door middel van een kleiner lettertype, is niet mogelijk. Het gebruik van stippellijnen om de cellen van elkaar te scheiden gaat ook niet. Het is wel mogelijk om de inhoud van een cel cursief, vet of onderstreept weer te geven.
De menubalk lijkt op het eerste gezicht een gewone GEM-menubalk te zijn. Bij het doorlopen van de pulldown-menu's (Wie weet hier een goed Nederlands woord voor?) verschijnen er ook pop-up-menu's. Door zorgvuldig navigeren met de muis kunnen de diverse commando's worden gekozen. Zoals gewoonlijk kunnen de commando's ook via het toetsenbord worden gegeven. Gemengd gebruik van muis en toetsenbord, zoals bij LDW-Power, lukt meestal niet. Het met de muis aangeven van 'ranges' gaat niet erg soepel. Zodra de muisknop losgelaten wordt, is geen wijziging van de range meer mogelijk. Het scrollen gaat overigens aanzienlijk sneller dan bij VIP-Professional.

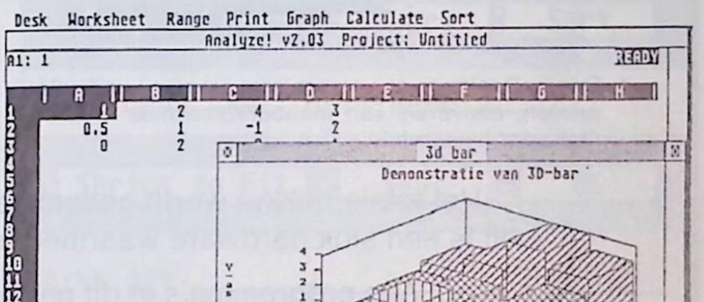
### De commando's

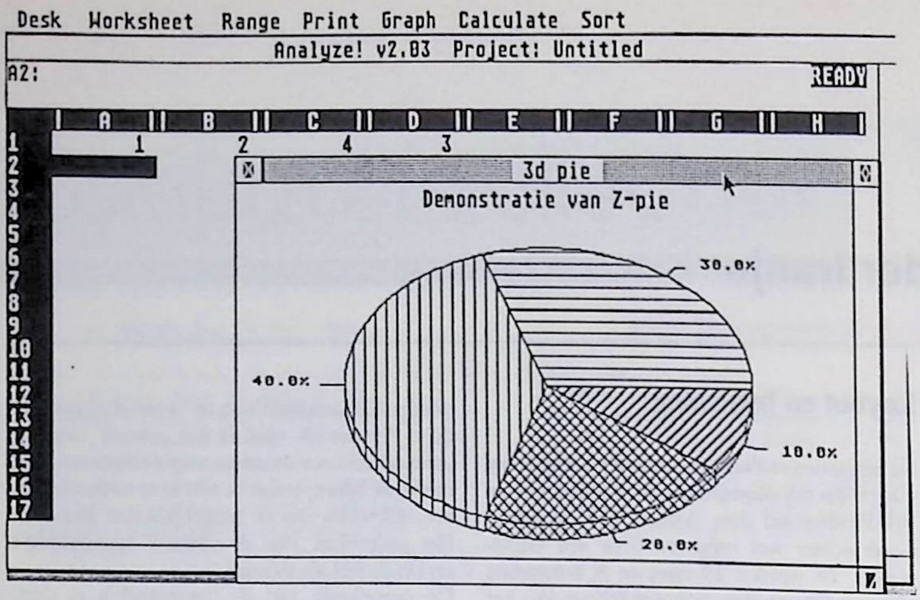
De commando's van Lotus 1-2-3 (versie 1) zijn merendeels ook in Analyze! aanwezig. Een belangrijke uitzondering is het ontbreken van de database-commando's en de 'what-if'-commando's. Persoonlijk vind ik het gebruik van een spreadsheet voor database-toepassingen sterk op misbruik lijken, zodat ik niet al te zwaar til aan het ontbreken van de mogelijkheden hiervoor. Het ontbreken van de 'what-if'-commando's ervaar ik wel als storend.
De organisatie van de commando's is sterk afwijkend (Kijk maar naar de menubalk). Nog afgezien van gewenningsproblemen betekent dit, dat de macro's van Lotus 1-2-3 vrijwel volledig onbruikbaar zijn. Daar staat tegenover dat Analyze! iets meer mogelijkheden biedt op grafisch gebied (3D-bar, Area en Z-pie). Grafieken kunnen rechtstreeks vanuit het programma worden afgedrukt op de printer (via GDOS). Het is in theorie zelfs mogelijk om een plotter te gebruiken, maar een 'plotter-driver' voor GDOS wordt niet meegeleverd. Titels en bijschriften in grafieken kunnen niet in een bijzondere stijl (vet, cursief e.d.) worden afgedrukt. Er wordt een speciale 'driver' voor de laserprinter meegeleverd. Deze 'driver' is nog niet door mij getest.

### De rekenfuncties

Vrijwel alle numerieke functies van Lotus 1-2-3 zijn aanwezig. De zogenaamde database-functies ontbreken echter (DCOUNT, DSUM e.d.). Het volledig ontbreken van de 'string'-functies laat het verschil met de meer geavanceerde spreadsheets, zoals Lotus 1-2-3 versie 2 en Symphony, echter pijnlijk voelen.
De rekennauwkeurigheid is goed (16 cijfers), maar het bereik van de getallen gaat niet bijzonder ver (8.5E37). Kennelijk worden de getallen niet in het IEEE-formaat opgeslagen.






## Compatibiliteit

Analyze! gebruikt normaal een eigen file-formaat. De files van Lotus 1-2-3 files (alleen versie 1) en VIP kunnen echter ook worden gelezen en geschreven. Door de afwijkende organisatie van de commando's zijn de macro's niet uitwisselbaar.

## Prestaties

De benchmarks van onze medewerker Berg Chabot laten zich helaas niet gebruiken voor het testen van Analyze! (o.a. door het ontbreken van 'Data Fill'). Ik heb daarom mijn toevlucht moeten nemen tot een iets eenvoudiger test. Een gebied van 100 regels bij 25 kolommen wordt gevuld met functies waarmee elke cel 1.001 maal de waarde van zijn linker buur krijgt. De testresultaten zijn:

Analyze! : 57 sec
LDW : 30 sec
VIP : 59 sec.

De diverse macro's staan op diskette ST19. De resultaten spreken voor zichzelf.

## Conclusie

Analyze! vormt een duidelijke verbetering ten opzichte van VIP-Professional. Het programma is echter te laat gekomen, want er is onderhand een nog veel beter spreadsheet verschenen (zie de bespreking van LDW-Power). Alleen als de prijs van Analyze! aanzienlijk verlaagd wordt, dan kan Analyze! wellicht nog meedoen. De extra mogelijkheden voor de grafieken worden anders wel duur betaald.

**Han Driesen**

*Producent: Micro-Systems Software, Inc.*
*Beschikbaar gesteld door: SoftPaquet*
*Prijs: fl 598,-*

# MultiDesk

## Een onmisbaar hulpmiddel?

**In elk nummer van ons blad vindt u besprekingen van softwareprodukten. We trachten de lezers door middel van deze besprekingen op de hoogte te houden van de ontwikkelingen. Meestal zijn dit programma's die reeds bestaan. Voor de redactie is dit vaak 'weer een tekstverwerker' of 'weer een spreadsheet'. Op de Cebit (de computerbeurs in Hannover) viel ons oog op een programma dat echt uniek is. Een accessory die het bedieningsgemak van de ST verveelvoudigt. Ondanks het feit dat het niet onze gewoonte is om produkten te bespreken die (nog) niet in Nederland verkrijgbaar zijn, willen we u dit niet onthouden.**

### De Desk Accessories

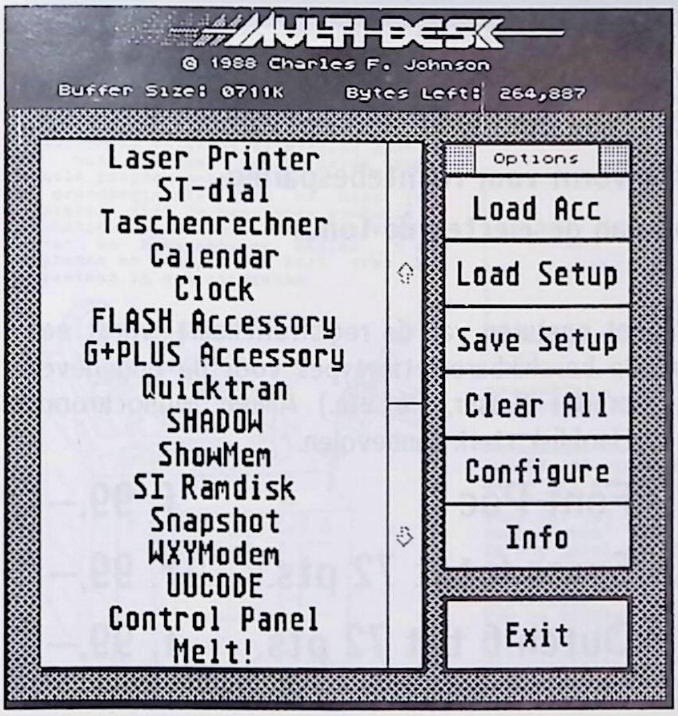
Iedereen die een paar weken een ST bezit, kent het gemak van accessories. Vanuit elk GEM-programma kunt u beschikken over alle mogelijke soorten hulpmiddelen: kladblokjes, rekenmachines, formatteerprogramma's, adresboekjes of zelfs spelletjes. Deze worden ingeladen tijdens het booten van de ST, zonder dat de gebruiker er iets voor hoeft te doen. Met een enkele muisklik zijn ze aan te roepen.

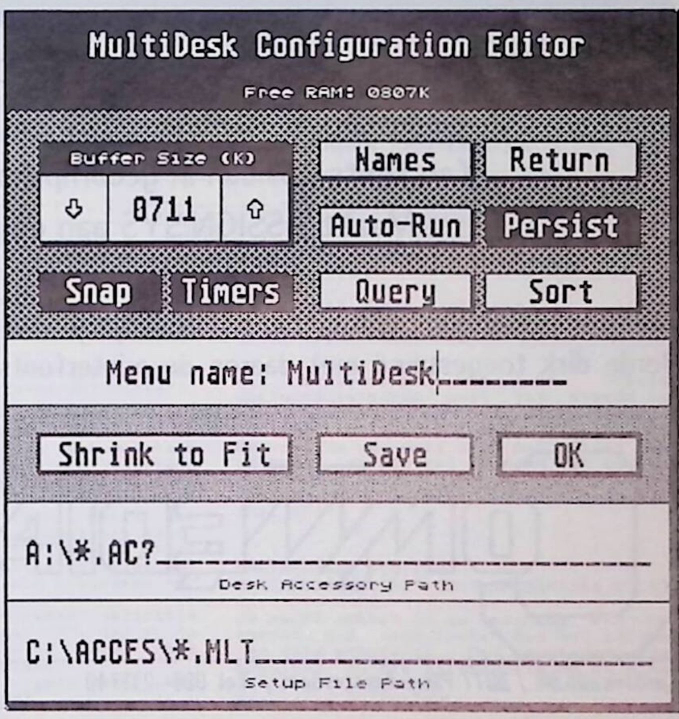
Iedereen die een paar maanden een ST bezit, kent de bezwaren van accessories. Het eerste bezwaar is de limiet van 6 accessories. Het tweede bezwaar is het geheugengebruik. Omdat accessories altijd in het geheugen aanwezig zijn is er minder geheugen beschikbaar voor uw programma. Een treffend voorbeeld is de Epson-emulator voor de laserprinter, die ongeveer een Megabyte aan geheugen reserveert. Zelfs voor een MEGA-2 is dit vaak te veel van het goede. Het derde bezwaar is de selectie van de accessories. In de Desktop blijken altijd de verkeerde accessories aanwezig te zijn. Zelfs programma's als Superboot (dat u accessories laat kiezen tijdens het booten) lossen dit probleem niet op. U dient toch altijd uw programma af te sluiten en opnieuw te booten.
Het zou dus ideaal zijn als we (meer dan 6) accessories konden laden zonder opnieuw te booten en deze ook weer verwijderen, waarbij het geheugen weer beschikbaar komt voor ons programma.

### MultiDesk

MultiDesk is geschreven door Charles F. Johnson en is afgeleid van het sharewareprogramma 'Desk-Manager'. Met Desk-Manager kunt u o.a. tijdens het booten kiezen welke accessories geladen moeten worden. MultiDesk gaat echter verder. Het is voor die gebruikers die reeds bekend zijn met het gebruik van accessories en deze ook regelmatig nodig hebben. Als de accessories in de hoofddirectory (root directory) staan, dan dienen hun extensies veranderd te worden. Ze hoeven echter niet in de hoofddirectory te staan. U kunt ze ook copiëren naar een folder of zelfs naar een andere drive. Als u twee diskdrives heeft, dan kunt u booten vanaf drive A, terwijl de accessories op drive B staan. Harddiskgebruikers kunnen de accessories op elke partitie plaatsen.

MultiDesk gebruikt standaard 90 Kbyte RAM, waarvan 64 Kbyte voor de accessory-buffer. Dat MultiDesk zelf maar 27 Kbyte gebruikt is te danken aan het feit dat het volledig in assembler geschreven is. De accesory-buffer op 2 Megabyte instellen (bij een MEGA ST4) is ook geen probleem. Na het laden van de accessories kan met de optie 'Shrink to Fit' de buffer op minimale grootte (de grootte die de ingeladen accessories samen nodig hebben) ingesteld worden.



## Het gebruik

Als u MultiDesk selecteert in de menubalk, dan ziet u een dialoogbox met 32 (jawel twee-en-dertig) accessory-slots. Met een simpele muisklik is elke accessory te selecteren. Als u aan 32 accessories niet genoeg heeft, dan is het mogelijk om in een van de slots MultiDesk te laden, waarmee dan weer 32 accessories geladen kunnen worden, waarvan dan weer een MultiDesk kan zijn, waarmee dan weer...etc.
Tijdens het testen kwam mijn Mega ST4 zonder geheugen te zitten, maar MultiDesk bleef zijn werk doen. Bijna elke willekeurige accessory is later in te laden. Ook is het mogelijk om bepaalde sets van accessories te definiëren en onder verschillende sets op te slaan. Met Load-Setup is het dan mogelijk om in een keer een set accessories in te laden.

Als de gebruiker alle accessories verwijdert en het buffer kleiner maakt, dan wordt het vrijgegeven geheugen weer netjes aan het systeem teruggegeven. Dit werkt echter alleen vanuit de desktop. Er zijn meer opties die uitsluitend werken vanuit de desktop. Als u ze gebruikt vanuit een programma, dan wordt u netjes getracteerd op een Alertbox.

Als u de extensie van MultiDesk van .ACC verandert in .PRG, dan kunt u het programma gewoon met de muis starten. U kunt dan uiteraard niet de accessories vanuit een programma aanroepen, maar u kunt wel accessories starten zonder opnieuw te hoeven booten.

## Geen nadelen?

Ik kon maar een paar nadelen in het gebruik van MultiDesk ervaren. Om een accessory te gebruiken moet twee keer met de muis geklikt worden; een keer op MultiDesk en een keer op de accessory zelf. Dit probleem is echter op verschillende manieren verminderd. In eerste instantie heeft de programmeur een soort van muissturing in het programma gebouwd. Op het moment dat u MultiDesk in de desktop selecteert, komt de dialoogbox van MultiDesk op het midden van het scherm tevoorschijn. Grote bewegingen met de muis zijn echter niet nodig, omdat de muis al gepositioneerd is op de eerste accessory in de selectiebox.

In het configuratie-menu kan de gebruiker kiezen voor de optie 'Persist'. Met deze optie ingeschakeld, wordt bij het aanklikken van MultiDesk de laatst gebruikte accessory aangeroepen zonder dat eerst het MultiDesk-menu verschijnt.

Verder werkt MultiDesk niet met alle accessories. Van de 100 accessories die ik getest heb, waren dit er echter maar twee: CPANEL++ en m'n eigen ST-DIAL (Hmm). Naast MultiDesk heeft de desktop echter nog 5 slots vrij waar deze 'ouderwets' geplaatst kunnen worden.

## Conclusie

MultiDesk is voor mensen die veel gebruik maken van verschillende accessories een ideaal hulpmiddel. Zeker harddisk-bezitters zullen dit programma weten te waarderen. Het bedieningsgemak en de vele opties maken het voor mij tot een inmiddels onmisbaar programma.

**Bert Rozenberg**

# Cursus APL

**Aflevering 4**

## Grondbeginselen Programmeren

**In deze aflevering leren we de grondbeginselen van het *PROGRAMMEREN.* Na afloop**

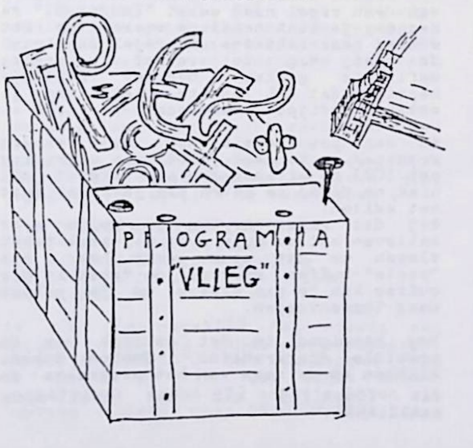
1. **kun je een programma definiëren en executeren.**
2. **kun je een programma editen.**
3. **kun je kommentaar in een programma toepassen.**
4. **kun je lussen programmeren met sprongopdracht en label.**
5. **kun je alternatieven programmeren met sprongopdracht en label.**
6. **weet je hoe je de uitvoering van een programma kunt onderbreken.**

Je hebt nu de *GEGEVENS* en de *KOMMANDO*'s goed leren kennen in afleveringen 2 en 3. Dan is het nu tijd om de *FUNKTIE*s te gaan bespreken.
Funkties zijn taal-elementen die iets DOEN, die gegevens bewerken.
De leukste funkties zijn de *PROGRAMMA*'s. Dat zijn funkties die je zelf maakt. Die programma's gaan we in dit en in de volgende aflevering behandelen.

Je zult zien dat programmeren niets anders is dan een of meer APL opdrachten tezamen opbergen in een kistje en daar een woord aan geven zodat je die opdrachten daarna enkel en alleen met dat ene woordje kunt aktiveren. Je hoeft dan niet elke keer alle opdrachten opnieuw over te typen. In wezen zul je dus eigenlijk weinig nieuws leren (nou ja, een beetje) want de hoofdzaak is hoe je zo'n programma-kistje maakt, verstart en gebruikt.

Tot nu toe hebben we APL funkties zoals + - × ÷ beschouwd als black-boxjes. We keken niet hoe de "+" precies werkte, zolang als het resultaat maar volgens de verwachting was.
Datzelfde geldt voor een programma als je dat eenmaal gemaakt hebt. Hoe de gedetailleerde inhoud van dat programma-kistje is interesseert je later niet meer, als je die kist maar kunt aktiveren door de naam op zijn etiketje te gebruiken en als dat programma dan maar werkt zoals beloofd. Zo'n programma wordt dus gekenmerkt door zijn NAAM.

Je zult eerst in deze aflevering de grondbeginselen van het programmeren leren; in de volgende aflevering gaan we ook wat pittiger zaken leren zoals "lokale programma-omgeving".
De grondbeginselen die we hier gaan bespreken zijn de begrippen exekutie en definitie, het editen van programma's, lussen en alternatieve akties in een programma en we spreken kort over het kommentaar in een programma.




### 4.1 Executie versus definitie

Het is belangrijk dat je onderscheid maakt tussen twee aspekten van het programmeren. Eerst ga je opgeven *WAT* er gedaan zal moeten worden, dat noemen we programma-*DEFINITIE*. Daarna kun je opdragen *DAT* de opdrachten van zo'n programma *UITGEVOERD* moeten worden, dat noemen we programma-*EXEKUTIE*.

Als voorbeeld van programma-*DEFINITIE* kijken we nog eens naar dat heel eenvoudige programmatje uit het begin. We definieerden dat als volgt:

```
     ∇HOY[CR] <=== naam van programma
[1]           <=== APL typt regelnummer
     ⎕←'OOK GOEIEMORGEN'[CR] <=== jij
                          typt opdracht
[2]           <=== APL typt volgende
                          regelnummer
     ∇ [CR] <=== jij vindt het welletjes
```

daarna is een voorbeeld van programma exekutie:

```
     HOY[CR] <=== programma aktiveren
OOK GOEIEMORGEN <=== programma antwoord
     HOY[CR] <=== en nog eens een keer
OOK GOEIEMORGEN
```

Dit illustreert de belangrijkste aspekten van programmeren in APL: definitie en exekutie, regelnummers en programma-naam. Daar gaan we eerst even wat uitvoeriger bij stil staan.

DEFINITIE TOESTAND
Je kunt grofweg zeggen dat APL in 3 toestanden kan zijn: 1) wachten op invoer, 2) uitvoeren van opdrachten (="exekutie") en 3) het opslaan van opdrachten (="programma-definitie"). Zie figuur.

Normaliter staat APL in de toestand "WACHT OP INVOER". In die toestand wacht APL net zolang op invoer totdat je een regel inbrengt.
Stel dat jij dan een regel geeft die begint met "∇" gevolgd door een naam. In ons geval was dat "∇*HOY*". Je komt dan in programma definitie en alles wat je verder intypt wordt als programma *HOY* opgeborgen.
Typ je tijdens die programma definitie een zin die eindigt met "∇", dan ga je uit die toestand en keer je terug naar de normale "wacht"-toestand.
Zie je hoe dit verhaal opgaat voor het ingevoerde programma *HOY*?

EXEKUTIE
In bovenstaande fig.4.1 zie je ook het alternatief van programma definitie namelijk *EXEKUTIE*. Dat gebeurt als je een regel invoert die niet met "∇" begint, bijvoorbeeld "2+3[CR]". APL voert dan die opdracht uit en keert na afloop weer terug naar de toestand "WACHT-OP-INVOER". Dus ook als je typt "*HOY*[CR]" (zonder "∇" ervoor). APL doet dan evenzo goed exekutie; maar nu van je programma "*HOY*".

REGELNUMMERS
Een volgend aspekt wat je met dit eenvoudige programma ziet zijn de regelnummers. Als je bovenstaande tekst letterlijk hebt ingetypt merkte je dat APL elke regel automatisch van een volgnummer voorzag. We zullen later zien dat we die nummers gebruiken om het programma te verbeteren.

NAAM
Meteen bij het openen van de programma-definitie geef je de naam van het programma. Dat is nodig omdat een werkruimte meerdere programma's kan bevatten. APL wil daarom weten welk programma je wilt definieren.
Zoals je ziet gebruik je die naam eveneens bij exekutie om te vertellen welk programma APL uit moet voeren.

Nu heb je een basiskennis van programma-definitie en van programma-exekutie. We gaan nu eerst heel even wat vertellen over kommentaar. Daarna pakken we de draad van programma-definite weer op met het veranderen van een bestaand programma (=programma-exekutie).

### 4.2 Commentaar

In APL kun je altijd verduidelijkende tekst toevoegen. Die tekst is er niet voor de komputer, die komputer slaat dat gewoon over. Zulke tekst is bedoeld om het programma voor jou of voor andere lezers te verduidelijken. We noemen zulke tekst: *KOMMENTAAR*.

APL herkent kommentaar aan het voorafgaande teken "⍝" (spreek uit "LAMP"). Laten we dat eens illustreren aan een nieuw programmatje.

```
+-----------+ ...[CR] +-----------+ ∇NAAM[CR] +-----------+
|           |<--------|   WACHT   |---------->| PROGRAMMA |
| EXEKUTIE  |         |    OP     |           | DEFINITIE |
|           |-------->|  INVOER   |<----------|           |
+-----------+  KLAAR  +-----------+  ...∇[CR] +-----------+
```

Fig.4.1: Drie APL toestanden (exekutie,wacht,definitie)

Als sekretaris van de Mokumse duivenmelkersklub moet jij steeds van elke binnengekomen duif de gevlogen afstand berekenen. Je krijgt dan hoeveel km een duif naar het zuiden was uitgezet en hoeveel km naar het Oosten; de afstand is de wortel van de som der kwadraten. Normaliter typ je voor elke duif in:

```
AFSTAND←400 300 [CR]
⎕←((AFSTAND[1]*2)+AFSTAND[2]*2)*0.5[CR]
```

Je maakt echter in de spanning van het moment erg veel fouten bij het intypen van deze expressie. Het is daarom veel handiger om die rekenregels op te slaan in een programma; we noemen dit programma DUIFVLIEG.

Daar gaan we

```
    ∇DUIFVLIEG[CR]
[1] ⍝⍝ Bereken afstand
    t.o.v. Amsterdam[CR]
[2] ⍝ Progr.: T. Rutten, 8 mei 86[CR]
[3] AFSTAND←400 300 ⍝ Startpunt [CR]
[4] ⎕←'De afstand is hemelsbreed:'[CR]
[5] ⎕←((AFSTAND[1]*2)+AFSTAND
    [2]*2)*0.5∇[CR]
```

Begrijp je deze 5 APL expressies? Je hebt in regel[3] en [5] de essentiele opdrachten ingetypt vergezeld van een nuttige tekst in regel[4].

Meteen even proberen. Typ in:

```
     DUIFVLIEG[CR] <=== en meteen komt:

De afstand is hemelsbreed:
500
```

Hoera, het werkt. Laten we nu dit programma eens nader bekijken.

1. In regel[1], regel[2] en rechts in regel[3] zie je het <u>KOMMENTAAR</u> dat begint met "⍝". In deze fase is dat nog niet zo nodig, maar als je later honderden programma's gemaakt hebt ben je blij als je aan zulk kommentaar snel ziet wat dit programma ook al weer doet.

2. Je ziet <u>OUTPUT</u> in [4] en [5]. Zie je het verschil tussen kommentaar en tekst-output? Output wordt tijdens exekutie WEL geprint, maar kommentaar NIET.

3. Je ziet ook weer het onderscheid tussen programma-<u>DEFINITIE</u> en programma-<u>EXEKUTIE</u> (eenvoudig intypen van "DUIFVLIEG[CR]").

Een valkuiltje is soms het sluiten van de programma-definitie met "∇". Als dat teken ∇ in het kommentaar valt, dan beschouwt APL dat ook als kommentaar en je programma-definitie eindigt dan niet. Geef in zo'n geval gewoon [CR] en zet je "∇" op de volgende regel.

## 4.3 Programma editen

Had je bovenstaand programma goed ingetypt? Ja? Dan heb je geluk gehad. Ikzelf maak in zoiets vaak foutjes. In dat geval is het nodig om het programma te verbeteren. We noemen dat met een mooi woord, het <u>EDITEN</u> van een programma.

We zullen vijf aspekten van het editen bekijken: het LISTEN van een programma, het VERWIJDEREN van een regel, het INVOEGEN van een regel, het VERPLAATSEN van een regel, en het VERBETEREN van een foutieve regel.

Wat je ook wilt, wissen, bekijken of wijzigen; het editen van een programma gaat altijd tijdens de programma-definitie. Dus altijd ∇naam.....editen...en.sluiten.met.∇.

### 4.3.1 Regels verbeteren

Het programma DUIFVLIEG is nu nog een beetje beperkt. Het rekent immers alleen uit wat de gevlogen afstand is als de duif op positie 400 300 gelost is. Dat kunnen we beter variabel maken, zodat je elke keer met een andere afstand kunt werken. Daartoe gaan we regel[3] waar AFSTAND een vaste waarde krijgt vervangen door een regel die om de getallen vraagt.

Dat wijzigen van een bestaande regel is eenvoudig. Je vraagt APL gewoon om die regel op het scherm te tonen en dan ga je die regel op het scherm veranderen. Dat tonen van een regel doe je met "[regelnr⎕]".

```
      ∇DUIFVLIEG[CR]  <===begin editen
[6]                 <===de komputer staat
                        klaar op regel[6]
                        naar wij zeggen:
    [3⎕] <===toon regel[3] voor editen
[3] AFSTAND←300 400 <=== dit typt APL
```

Je gaat nu met de kursor over de regel heen en gebruikt zonodig de [DEL] en [INS] tot je de volgende tekst op het scherm hebt staan:

```
[3] AFSTAND←⎕ ⍝ afstand inlezen [CR]
```

Daarna nog "∇[CR]" als je klaar bent met editen.

Even uitproberen:

```
     DUIFVLIEG[CR]     en meteen komt:

⎕:           <=== APL vraagt om invoer
    300 400 [CR]     <===jouw antwoord
De afstand is hemelsbreed:
500
```

Als je veel fouten in een zin hebt gemaakt kun je die zin ook eenvoudigweg in zijn geheel overtypen. Bijvoorbeeld:

```
∇DUIFVLIEG[3] AFSTAND←⎕
⍝ afstand inlezen[CR]
[4] <=== APL geeft volgende regelnummer
    ∇ ## <=== jij bent klaar
```

### 4.3.2 Regels invoeren

Nu zullen we het INVOEGEN van regels illustreren.

Tijdens de werking van het programma DUIFVLIEG komt de vraag om input ("⎕:") zomaar uit de lucht vallen. Dat kunnen we verduidelijken door tevoren een verklarende tekst uit te laten printen. Daartoe gaan we tussen regel [2] en [3] een opdracht tussenvoegen. We gebruiken dan een <u>gebroken regelnummer</u>.

Voorbeeld

```
    ∇DUIFVLIEG[CR] <=== begin editen
[6]        <===de komputer staat klaar
                op [6], maar wij zeggen:
[2.1]'Geef 2 getallen:
km. ten Zuiden '[CR]
[2.2]      <===nu zegt APL zelf "[2.2]"
    'en km. ten Oosten v.Amsterdam'[CR]
[2.3] ∇ [CR]               <===klaar
```

Even uitproberen:

```
    DUIFVLIEG[CR] <=== en meteen komt:
Geef 2 getallen:  km.  ten Zuiden
en km.  ten Oosten v.Amsterdam
⎕:
    300 400 [CR] <===jouw antwoord

De afstand is hemelsbreed:
500
```

### 4.3.3 Programma listen

Nu zullen we het LISTEN van het programma illustreren.

Hoe ziet het programma er nu uiteindelijk uit? We typen daartoe:

```
    ∇DUIFVLIEG[⎕]∇[CR]
```

APL geeft nu:

```
[0] DUIFVLIEG
[1] ⍝⍝ Bereken afstand t.o.v. Amsterdam
[2] ⍝ Progr. T. Rutten, 8 mei 86
[3] 'Geef 2 getallen: km. ten Zuiden '
[4] 'en km. ten Oosten v.Amsterdam'
[5] AFSTAND←⎕
[6] ⎕←'De afstand is hemelsbreed:'
[7] ⎕←((AFSTAND[1]*2)+AFSTAND[2]*2)*0.5
```

Zie je hoe het programma is geworden? Alles is nu weer netjes doorgenummerd en de ingevoegde regels [2.1] en [2.2] zijn daardoor veranderd in [3] en [4]. Dat gebeurt automatisch zodra je de programma-definitie sluit (zolang je nog geen programma-definitie sluit verandert er niets aan die nummering!).

Dit was 1 manier om de tekst van je programma te zien. We noemen dat het <u>listen</u> van je programma.
We hoeven die opdracht tot listen niet in 1 regel te doen. Je kunt ook eerst de programma-definitie aanvangen, en dan pas later om een listing vragen, en pas daarna (zo je dat wilt) de programma-definitie beeindigen.

Voorbeeld

```
    ∇DUIFVLIEG[CR]
[6]
     [⎕][CR]
.....  <=== nu volgt de listing
[6] ∇[CR] <=== nu eindigt de definitie
```

Op dit listings-kommando bestaan de volgende varianten:

1. [⎕] list komplete programma
2. [⎕5] list programma te beginnen met regel 5
3. [8⎕] list uitsluitend regel 8

### 4.3.4 Regels verwijderen

Een regel verwijder je met de opdracht "[∆nummer]". Voor nummer geef je het volgnummer van de regel die je kwijt wilt.

Ook dit kun je weer in 1 opdracht doen:

```
     ∇DUIFVLIEG[∆3]∇[CR]
```

of je kunt het in stapjes doen:

```
    ∇DUIFVLIEG[CR] <=== start definitie
[7]                <=== dit typt APL
    [∆3][CR] <=== jij zegt regel 3 weg
[3] ... <=== Nog iets anders doen?
             Bijvoorbeeld [⎕]?
    ∇[CR] <=== definitie beeindigen.
```

Er is een belangrijk verschil tussen de snelle en de stapsgewijze manier. Zodra je namelijk de definitie met "∇" beeindigt zal APL alle regels hernummeren. Als je regel[3] hebt gewist, dan krijgt de vroegere regel[4] volgnummer [3] en zo schuift alles op. Maar bij de stapsgewijze uitvoering is dus nog niet hernummerd omdat je nog geen ∇ hebt gegeven. Nummer [3] is dan nog steeds vrij en regel[4] heeft nog steeds nummer [4]. Dat verandert pas na "∇".

Als je meerdere regels wilt wissen kun je meerdere nummers tegelijkertijd opgeven; bijvoorbeeld:

```
     ∇DUIFVLIEG[∆2 3 4]∇
```

### 4.3.5 Regels verplaatsen

Om een regel te verplaatsen gebruik je de mogelijkheden van zonet. We verplaatsen namelijk door het volgnummer van de betreffende regel te wijzigen. Stel bijvoorbeeld dat je regel[2] eigenlijk tussen regel[6] en regel[7] had willen hebben (niet erg gebruikelijk!). Je zegt dan

```
    ∇DUIFVLIEG[2⎕]
[2] ⍝ Progr.  T.  Rutten, 8 mei 86
```

Nu verander je de "[2]" in "[6.1]" en drukt op [CR]. Dus:

```
[6.1] ⍝ Progr.  T.  Rutten, 8 mei 86
```

Als je het programma nu uitlijst, zie je dat deze regel nu twee keer voorkomt. Allereerst tussen regel[6] en [7], en ook nog steeds als regel[2]. Kijk maar

```
    [⎕][CR]
[0] DUIFVLIEG
[1] ⍝⍝ Bereken afstand t.o.v. Amsterdam
[2] ⍝ Progr. T. Rutten, 8 mei 86
[3] 'Geef 2 getallen: km. ten Zuiden '
[4] 'en km. ten Oosten v.Amsterdam'
[5] AFSTAND←⎕ ⍝ afstand inlezen
[6] ⎕←'De afstand is hemelsbreed:'
[6.1] ⍝ Progr. T. Rutten, 8 mei 86
[7] ⎕←((AFSTAND[1]*2)+AFSTAND[2]*2)*0.5
```

Het enige wat je nu dus nog moet doen is die overbodige regel[2] wissen. Daartoe typ je nog:

```
     [∆2] ∇[CR]
```

### 4.3.6 Gemakkelijker Atari-mogelijkheden

Bovenstaande uiteenzetting gaf het standaard editen in APL. Er zijn echter op de Atari nog meer, en veel gemakkelijker, mogelijkheden.

Ten eerste hoef je voor het modificeren van een regel niet eerst "[nummer⎕]" te zeggen; je kunt namelijk overal op het scherm naar behoefte een regel wijzigen. Je moet dan wel eerst programma definitie gestart hebben, want je begrijpt dat APL anders niets van je akties begrijpt! Dus eerst

∇naam

en dan pas regels op het scherm wijzigen. Vergeet niet die wijziging met [CR] te effektueren en vergeet ook niet om ∇[CR] te geven als je klaar bent met editen.
Bij dat wijzigen kun je zelfs naar believen met de muis over een stuk tekst slepen en dat stuk tekst naar een "paste"-buffer sturen; de tekst in die buffer kun je dan elders op je scherm weer tussenvoegen.

Nog handiger is het gebruik van de speciale Atari-editor. Gewoon dubbel klikken op de naam van het programma en dan [Open fn] uit het [Edit]-menu aanklikken.

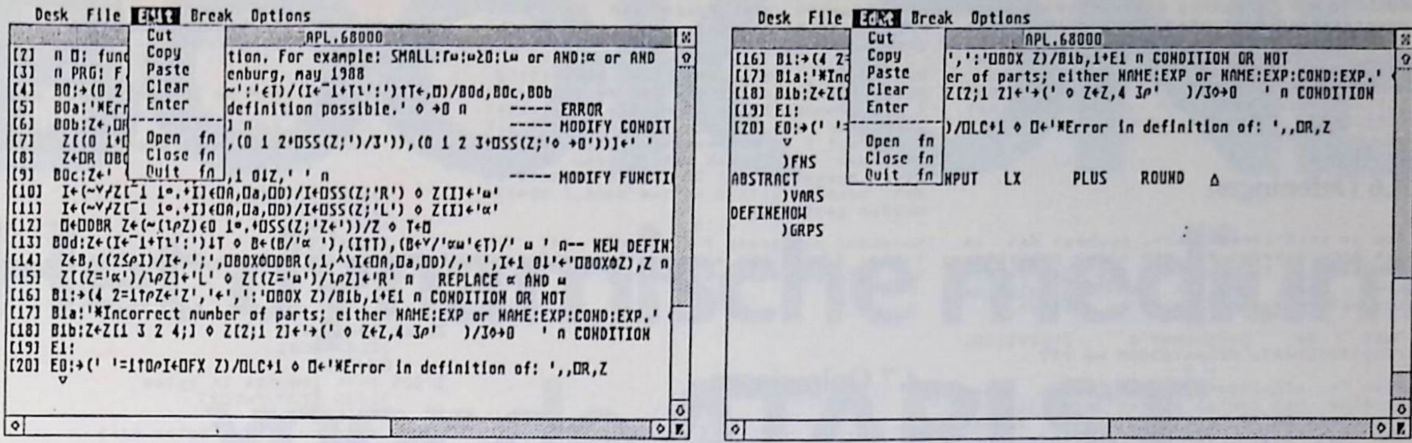



Fig.4.2: Opties in Edit-menu.

## 4.4 Sprongen en labels

De normale volgorde waarop APL de opdrachten uitvoert is: een voor een, van boven naar beneden. In deze paragraaf laten we zien dat je die volgorde kunt wijzigen met de sprongopdracht.
Als je APL met zo'n sprongopdracht terug stuurt naar een reeds eerder uitgevoerde opdracht, dan maak je een lus. Als je daarentegen vooruit springt, over opdrachten heen, dan programmeer je zogenaamde alternatieven. Beide gevallen gaan we in deze paragraaf bespreken.

### 4.4.1 Sprong

Op het punt vanwaar je wilt afwijken van de normale volgorde zet je de sprongopdracht. Die begint met "→" (spreek uit SPRING NAAR). Daarachter geef je op waarheen je wilt springen. Het punt waarheen je wilt springen markeer je met een zogenaamde label. Dit is een zelf te verzinnen woordje (net als de gewone variabele) dat gekoppeld is aan een regelnummer. Dat koppelen doe je door dat woord aan het begin van een programmaregel te schrijven, gevolgd door een ":".
Bij de sprong verwijs je naar die regel door achter de "→" dat woordje te zetten. We zullen dat eens voor doen aan de hand van een lus.

### 4.4.2 Lus

Ik ga het volgende programma *FRUITAUTOMAAT* maken:

```
    ∇FRUITAUTOMAAT[CR]
[1] ⍝⍝ Simuleer fruit automaat[CR]
[2] ⍝ PRG: Eke v.Batenburg DD.
      1 april 88[CR]
[3] FRUIT←4 6⍴'APPEL PEER  KERS
      BANAAN'[CR]
[4] FRUIT←FRUIT[1 2 2 2 2 3 3 3 3 3
      4 4 4 4 4 4;][CR]
[5] DRAAIOPNIEUW: ⍝
      let op de dubbel punt !!!!![CR]
[6] ⎕←FRUIT[?15;][CR]
[7] →DRAAIOPNIEUW[CR]
[8] ∇[CR]
```

In regel[5] heb ik het woordje *DRAAIOPNIEUW* gekoppeld aan die regel. Met de pijl "→" in regel[7] draag ik APL op om steeds weer terug te gaan naar regel[5].

Je zult merken dat dit programma oneindig lang door blijft gaan. Het zou daarom veel mooier zijn als dit programma stopt wanneer je een APPEL hebt. Dat programmeren we even door regel[6] en regel[7] te veranderen in:

```
    ∇FRUITAUTOMAAT[CR]
[6] ⎕←FRUIT[NR←?15;][CR]
[7] →(NR≠1)/DRAAIOPNIEUW[CR]
[8] 'HOERA, DE JACKPOT!!!!'[CR]
[9] ∇[CR]
```

Je ziet in regel[7] nog steeds een sprong, maar nu onder een voorwaarde, namelijk "(NR≠1)". Dat wil zeggen "als we niet de 1e uit fruit hebben getrokken". Regel [7] moet je lezen als "spring alleen naar *DRAAIOPNIEUW* terug als variabele *NR* ongelijk aan 1 is, en ga anders gewoon door". (*2)

---

(*2) Later zul je zien dat "(expressie)/label" een gewone APL opdracht is zonder magie. Als de expressie "waar" is staat er "1/label" (spreek uit: 1 REPLICEER LABEL) en dat geeft op zijn beurt "label"; als de expressie "onwaar" is dan staat er "0/label" en dat geeft geen getal (een zogenaamde lege vektor). Dit alles leggen we pas in aflevering 8 uit, maar nu kun je zo'n konstruktie best gebruiken zonder hem te begrijpen.

---

### 4.4.3 Onderbreken

Nu je lussen kunt maken kun je jezelf ook in problemen brengen door een eeuwig durend programma te aktiveren. Kijk maar eens naar onze allereerste versie van programma FRUITAUTOMAAT. Wat moet je doen als zo'n programma onverhoopt steeds maar dezelfde regels uitvoert? Of wat moet je doen als een programma naar je zin te lang duurt en je wilt weer terug naar de "wacht-op-invoer" status?

1. een programma dat steeds maar output geeft kun je tijdelijk *ONDERBREKEN* door [CTL/S] (toets [CTL] ingedrukt houden en dan op de [S] drukken) en weer hervatten met [CTL/Q].
2. Definitief stoppen doe je met [CTL/ESC].

### 4.4.4 Alternatieven

Het komt ook vaak voor dat je enige programma-opdrachten hebt die niet altijd mogen gebeuren. Stel bijvoorbeeld dat je uit wilt printen KAMPIOEN als iemand de 500 meter sneller schaatst dan 38 sekonden. Maar doet hij er langer over dan mag dat niet geprint worden. Zo'n programma ziet er als volgt uit:

```
    ∇SCHAATS500M[CR]
[1] ⍝⍝ Prijs als sneller dan 38 sek.[CR]
[2] ⍝ PRG: E.v.Batenburg, dd.28/12/88[CR]
[3] ⍞←'Tijd in sekonden:'[CR]
[4] TIJD←⎕ ⍝ lees gereden tijd in [CR]
[5] B1:→(TIJD>38)/E1 ⍝ spring zonodig
      over regel[6][CR]
[6] ⎕←'KAMPIOEN'[CR]
[7] E1: ∇[CR]
```

Soms heb je te maken met meer dan 1 alternatief. Bijvoorbeeld als je een programma PTT wilt maken dat vertelt hoeveel je moet frankeren: f.075 als de brief minder dan 20 gram weegt, f1.50 tot aan 50 gram, en anders f.3.00. Zo'n programma met alternatieven ziet er in schema als volgt uit:

Fig.4.2:

Schema 3 alternatieven

```
+------------------------+
|  \GEWICHT     /        |
|    \        /          |
|     \      /           |
| 0-20\  /20-50 |>50     |
| -----+--------+-----   |
| P←75 | P←150  | P←300  |
+------------------------+
```

Het bijbehorende programma ziet er als volgt uit:

```
     ∇PTT [CR]
[1]  ⍝⍝ bereken frankering afhankelijk
       van gewicht R[CR]
[2]  ⍝ GEWICHT: briefgewicht (gram) [CR]
[3]  ⍝ PORT: frankering (centen) [CR]
[4]  ⍝ PRG: Eke v.Batenburg, DD.jan.1988[CR]
[5]  →(GEWICHT>50)/VEEL[CR] >-----------+
[6]  →(GEWICHT>20)/GEMIDDELD[CR] >--+   |
[7]  KLEIN: PORT←75[CR]             |   |
[8]  →END[CR] >---------------------|---|-+
[9]  GEMIDDELD: PORT←150[CR]  <-----+   | |
[10] →END[CR] >-------------------------|-+
[11] VEEL: PORT←300[CR]  <--------------+ |
[12] END:[CR] <---------------------------++
                                      (*3)
[13] ∇
```

Zie je dat ik in regel[7] een label *KLEIN* gebruik waar ik niet heen spring? Dat mag; ik gebruik die label hier om duidelijk de 3 alternatieven KLEIN, GEMIDDELD en VEEL aan te geven.

## 4.5 Afsluiting

Je hebt nu de belangrijkste kenmerken van programmeren leren kennen.
Je weet nu het onderscheid tussen programma-*DEFINITIE* en *EXEKUTIE*.
Je hebt een programma leren editen met de volgende kommando's:

| | |
|---|---|
| ∇*NAAM* | Start definitie van programma "*NAAM*" |
| [⎕] | List programma |
| [⎕*N*] | List vanaf regel *N* |
| [*N*⎕] | Toon regel *N* voor wijziging |
| [∆*N*] | Wis regel *N* |
| [*N.M*]... | Voeg "..." toe achter regel *N* |
| [*N*]... | Vervang regel *N* door andere tekst |

Fig.4.3: Edit kommando's

Verder heb je de *LABEL* leren kennen om een regel te markeren; je zette zo'n label vooraan in de regel gevolgd door een dubbele punt. Je hebt die label leren gebruiken met de sprong.
Voor wat betreft de *SPRONG* (met "→") heb je de konditionele sprong leren kennen "→(*KONDITIE*)/*LABEL*". Je hebt gezien hoe je met de sprong lussen kunt maken door terug te springen. Ook heb je gezien

---

(*3) Je ziet dat zo'n simpel probleem een akelig onoverzichtelijk programma geeft. Veel APL-ers houden daarom helemaal niet van sprongen. Ikzelf zou dit probleem bijvoorbeeld programmeren als:

*PORT←75 150 300[1+(GEWICHT>50)+GEWICHT>20][CR]*

---

hoe je met de sprong alternatieven kunt programmeren door over opdrachten heen te springen.
Tenslotte heb je ook gezien hoe je een werkend programma tijdelijk kunt onderbreken met [CTL/S] (hervatten met [CTL/Q]) en permanent onderbreken met [CTL/ESC].

## 4.6 Oefeningen

Als je zelf ideeen hebt, probeer dan om het programmeren met een eigen projektje in praktijk te brengen. Als je zoiets niet hebt, gebruik dan de volgende suggesties om de theorie te oefenen.

\- 1 -

Maak de programma's DUIFVLIEG, FRUITAUTOMAAT, SCHAATS500M en PTT.

\- 2 -

Toen je afgelopen winter griep kreeg groeide het aantal zieken dagelijks 5%. Variabele GRIEP bevat het aantal zieken aan het begin van de epidemie, het dagelijkse groeipercentage (in dit geval dus 0.05), en het aantal dagen dat je de griep epidemie wilt simuleren.
Maak een programma EPIDEMIE dat de hoeveel zieken dagelijks berekent en print.

\- 3 -

In variabele GEGEVENS staan alle cijfers die je dit jaar behaald hebt. Maak een programma ANALYSE dat van de variabele GEGEVENS het minimum getal (gebruik "∆gegevens"), het grootste getal en het totaal aantal getallen berekent en dat met een net tekstje uit print.

\- 4 -

Vul dit programma aan zodat het ook de mediaan berekent. De mediaan is een maat voor het gemiddelde; je sorteert daartoe de getallen eerst en neemt dan het middelste getal. Neem als mediaan bij een even aantal getallen het gemiddelde van de middelste twee van alle gesorteerde getallen.

\- 5 -

Op de Bretonse kusten groeit een wier. Dat bedekt oorspronkelijk 30 vierkante meter.
Op die rotsen zitten ook slakken (ongeveer een kilo) die van dat wier eten. Door dat eten neemt het wieroppervlak af, maar de slakken nemen toe. Gelukkig neemt het wieroppervlak tegelijkertijd toe door de stralende zon. De toe- en afname van beiden is dagelijks volgens de formules:

```
WIERTOENAME=0.01xWIERxWIER - 0.2xWIERxSLAK
SLAKTOENAME=
-0.01xSLAKxSLAK + 0.1x0.2xWIERxSLAK
```

Laat het verloop van slakken en wieren gedurende 1 maand dagelijks printen door programma SLAKWIER.

\- 6 -

Maak een programma COVCOR dat een zogenaamde "KOVARIANTIE"-matrix COV omzet naar een "korrelatie"-matrix COR. Daartoe deel je de 1e regel en 1e kolom van COV door diens 1e diagonaalelement (dat wil zeggen door element [1;1]); de 2e regel en 2e kolom door het 2e diagonaal element, enzovoorts.

\- 7 -

In vektor ACTS staan honderden getallen die aangeven welke handelingen een stekelbaars heeft uitgevoerd. Bijvoorbeeld: 1 2 3 2 3 2 1 3 2 5 3 1.
In variabele HYPTHSE staan 2 handelingen waarvan je vermoedt dat die elkaar regelmatig opvolgen; bijvoorbeeld 3 1.
Maak een programma SEKWENTIE die in variabele Z twee getallen zet: hoeveel keer HYPTHSE[1] gevolgd werd door HYPTHSE[2] en hoeveel keer HYPTHSE[1] door een andere handeling werd gevolgd.

\- 8 -

Nu gaan we het vorige programma een beetje veralgemeniseren. In variabele ACTS staan honderden letters die aangeven welke handelingen een bioloog waarnam bij een pit-bull terrier (E=eten, S=slapen, B=bijten, G=grommen, enzovoorts). Bijvoorbeeld:

```
      ACTS←'GBEESBBEEBBSG']{CR]
```

Maak een programma SEQUENCE dat ACTS bekijkt en voor de letters in variabele VRG (bijvoorbeeld de letters "BES" voor Bijten, Eten, Slapen) een vierkante tabel terug geeft. Op regel 3 kolom 2 van die tabel staat hoeveel keer die hond handeling 3 liet volgen door handeling 2. Dus:

```
      SEQUENCE[CR] geeft
2 2 1 <=== B gevolgd door B E S
1 2 0 <=== E gevolgd door B E S
1 0 0 <=== S gevolgd door B E S
```

Maak ook zelf eens zo'n ethogram (een rij letters) van je zusje, je vriendje of je moeder en kijk welke handelingen het meeste voorafgaan aan het boos worden; en welke daar weer aan vooraf gaan.

\- 9 -

Een pittig programma ter afsluiting. Maak het programma BOX dat de tekst-rij V met woorden bij elke spatie in stukken knipt en een tabel T levert met elk woord uit V in een aparte regel. Je mag er voor het gemak van uitgaan dat elk woord hoogstens uit 20 tekens bestaat en dat woorden altijd netjes door 1 spatie worden gescheiden.

\- 10 -

Verander programma BOX zodanig dat die vektor V niet bij elke spatie knipt maar bij het voorkomen van "V[1]".

## 4.7 Oplossingen

\- 1 -

```
      VANCE←'/The face/The green
pearl/Maske'[CR]
      ASIMOV←'/Mysteries/Authorised
Murder'[CR]

CHALKER←'/Soulrider/Changewinds'[CR]
      CHRISTIE←'/Haloween/Hercule
Poirot'[CR]
      )GROUP SF VANCE CHALKER ASIMOV[CR]
      )GROUP DETECTIVE ASIMOV
CHRISTIE[CR]
      )GRP SF[CR]
VANCE CHALKER ASIMOV
      )GRP DETECTIVE[CR]
ASIMOV CHRISTIE
```

\- 2 -

```
      )SAVE GROEPEN[CR]
      )CLEAR[CR]
```

Nu geven )GRPS[CR] )FNS[CR] en )VARS[CR] allemaal aan dat er niets in de AW is.

```
      )PCOPY GROEPEN SF[CR]
      )PCOPY GROEPEN DETECTIVE[CR]
```

Kontroleren:

```
      )GRPS[CR]
SF DETECTIVE
      )GRP SF[CR]
VANCE CHALKER ASIMOV
      )GRP DETECTIVE[CR]
ASIMOV CHRISTIE
```

\- 3 -

Groep wissen (struktuur en objekten):

```
      )ERASE DETECTIVE[CR]
```

Kontrole:

```
      )VARS[CR]
CHALKER VANCE <=== Nu is ook ASIMOV weg!
      )GRP SF[CR]
ASIMOV CHALKER VANCE <=== maar ASIMOV
staat toch nog WEL in de groepsstruktuur
```

\- 4 -

```
      )GROUP SF[CR]
      )GRP SF[CR]
      <=== leeg
      )VARS[CR]
CHALKER VANCE <=== objekten zelf nog wel
aanwezig
```

\- 5 -

Eerst leegmaken

```
      )CLEAR[CR]
      ⎕WA[CR]
31608 <=== grootte in bytes
      )LOAD STGRAPH[CR]
      ⎕WA[CR]
26600 <=== nu minder bytes vrij
```

\- 6 -

Eerst laden:

```
      )LOAD STGRAPH[CR]
```

Onder andere naam saven:

```
      )SAVE KOPIETJE[CR]
```

\- 7 -

Eerst laden:

```
      )LOAD KOPIETJE[CR]
```

Dan objekt in AW wissen:

```
      )ERASE POLYLINE[CR]
```

Nieuwe toestand saven:

```
      )SAVE KOPIETJE[CR]
```

\- 8 -

```
      )DROP KOPIETJE[CR]
```

\- 9 -

```
)LOAD STGRAPH[CR]
      ∧)FNS[CR]
...POLYLINE ∆WIN...
      )ERASE POLYLINE BESTAATNIET[CR]
      )PCOPY STGRAPH POLYLINE ∆WIN
ONBEKEND[CR]
```

\- 10 -

```
      INTRO←'Hallo, daar gaan we dan'[CR]
      ⎕LX←INTRO[CR]
      )SAVE MIJNWERK[CR]
      )LOAD MIJNWERK[CR]
```

**Eke van Batenburg**

# Nieuws uit onderwijsland

**Om de toch al zwaar belaste onderwijsmensen tegemoet te komen, hebben software-buro's min of meer landelijk besloten om onderwijs-administratiepakketten te lanceren. Het is, alsof ze het afgesproken hebben, want in een periode van nog geen twee maanden zijn deze mensen bezig om vijf verschillende administratie-pakketten op de markt te introduceren.**

Eindelijk kan de leerkracht nogal wat gegevens in de computer stoppen, om meer tijd vrij te maken voor het onderwijsgebeuren. Ook directeuren kunnen door dit stuk gemak, tijd winnen om de administratieve investeringen te ontvluchten ten behoeve van onderwijsbegeleiding. In de praktijk van alle dag blijkt maar al te vaak, dat staf-en directie functies devalueren tot veredeld administratief werk. Met de komst van de schooladministratie-pakketten wordt in ieder geval een mogelijkheid geschapen om daar voor een deel een einde aan te maken. Nieuw in deze branche is, dat ex-onderwijsmensen steeds meer overgaan tot programmeren. Daar, waar deze onderwijsmensen daadwerkelijk erin slagen om een programmeertaal als instrument te gebruiken in het bestrijden van de tekortkomingen, mogen wij juichend een flink gat in de lucht springen. Zij weten immers waar het om gaat, waar behoefte aan is in het onderwijsveld, hoe de toekomstige gebruiker ermee om zal gaan. Heb je eenmaal het krijt geroken, dan beschik je over een kostbare ervaring aan informatie, die beginpunt kan zijn van een schitterend pakket. Gaat de onderwijsman een stuk hand in hand mee met de beroepsprogrammeur, dan spreken we over pakketten, waar half Nederland op zit te wachten. Doet de onderwijsman of vrouw het alleen, dan zal menige programmeur regelmatig zijn wenkbrauwen moeten fronsen. Kortom, we gaan toch de goede kant uit na jaren de kat uit de boom gekeken te hebben. De gemeente Utrecht doet er goed aan, al deze pakketten op zicht te vragen en er iets goeds van te laten maken voor alle basisscholen. Het begin is er al, nu moet nog een serieuze uitgever de draad oppakken.

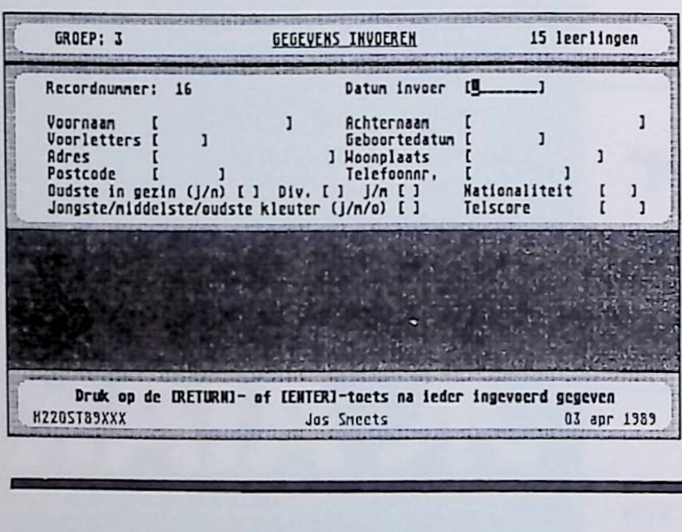
## Iris

We beginnen met IRIS, een modulair en menu-gestuurd Leerlingen administratie programma voor de ATARI ST. Op verzoek van ATARI BENELUX is dit pakket ontwikkeld door Iris Produkties in Barneveld. De verkoop en verspreiding geschiedt door een echte onderwijsuitgever, namelijk Heutink Primair Onderwijs te Rijssen. Iris Produkties heeft het pakket voor zowel de MS-DOS, de BBC Master 512 als voor de ATARI ST computer geschreven. M.b.v. een emulator moet het zelfs draaien op de AMIGA 500 als op de BBC Archimedes. Wie zei ooit eens, dat pluriformiteit niet te betalen was?
Aan de ATARI ST-versie is te zien, dat de programmeurs normaliter een MS-DOS-scherm voor zich hebben. Men denkt, dat alles met het toetsenbord moet, maar het kan nog net met de muis. Grafisch mist het derhalve dan ook de charme die de ST-pakketten doorgaans uitstralen. Het programma wordt via verschillende toegangscodes bewaakt. Men krijgt een startcode om in het programma te komen en drie verschillende andere codes om leerlingengegevens, lerarengegevens en de overgang te bekijken. De afsluitbare dossierkast wordt inderdaad vervangen door een code of een toegangswoord.
Iris Produkties speelt in op de toekomst. Mocht de Basisschool na 1992 een MS-DOS in de maag gesplitst krijgen van de overheid, dan kunnen alle gegevens zonder problemen in de MS-DOS-versie gebruikt worden. We noemen hieronder enkele van de mogelijkheden: naast een persoonsbestand zit er een schooloverzicht in, met telopties, gewicht van leerlingen, formatie- plaatsen enz. De printer kan records met koppen uitprinten, verjaardagslijsten, acceptgiro's etc. Het pakket kost f 199,- Voor informatie: 03426-2091

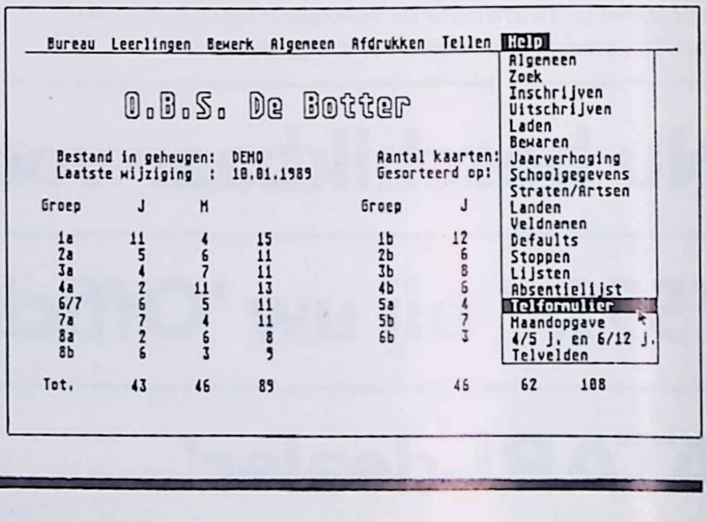
## Compudress

Compudress Educatief heeft SYSTRON een schitterend administratie-pakket laten ontwikkelen, dat wel zo aardig is opgebouwd, dat een gelijkenis met HYPERCARD van de MAC meteen opvalt.
Ook hier zijn toegangen noodzakelijk om verschillende nivo's van het programma te bereiken. Het voordeel van deze toegangen is, dat de gebruiker van het pakket elk bestand vooraf op drie lagen kan beveiligen middels zelf in te geven toegangscodes bestaande uit 4 cijfers. Helaas zijn hier geen letters in opgenomen, waardoor die volhouder in het weekend inderdaad na maximaal 9998 keer proberen alle gegevens voor zich kan opeisen. Maar U snapt het wel, de net wat te nieuwsgierige ouder, of de wat rondlummelende leerling moet een lotje uit de loterij hebben, wil

*IRIS*




*H.v.d.Keur: Leerlingenadministratie*



deze tot de gegevens komen. Nu kan iedere kollega in elk bestand totaal verschillende codes hanteren, waardoor een systeembeheerder niet alle informatie kan aanboren. Het enorme grote voordeel van deze beveiliging is, dat nu ouders gerust met het pakket kunnen omgaan. De lagere toegang staat hun wel toe verjaardagsuitdraaien te maken, maar niet de medische achtergrondgegevens. Eindelijk hoeft de directeur niet over de schouders heen te kijken, of men niet al te veel gegevens uit de computer zuigt.
In het programma komt men overzichtelijke schermen tegen met op vaste plaatsen duidelijke BUTTONS, die ondergetekende doen denken aan programma's uit de diepteprojekten. Elke BUTTON brengt de gebruiker op een manier, die goed te volgen is, naar een andere kaart. Terug klikken naar een willekeurige kaart met gegevens is te allen tijde mogelijk. In een apart onderdeel komen wij dan terecht in het afdruksysteem. De meeste wensen, om gegevens af te drukken, worden hier gehonoreerd met een optie. Het pakket wordt door Compudress Educatief op naam gesteld, waardoor alle uitdraaien een eigen kop van de school krijgen. Gelukkig omzeilt men de kopieerbeveiliging steeds vaker met dit soort sympathieke oplossingen. Het pakket moet f 199,- kosten en zal binnenkort geleverd worden. Voor inlichtingen: Compudress Educatief, tel: 03481-2124

## Leerlingenadministratie Basisschool

Henk van de Keur uit Ridderkerk heeft voor de school waar hij werkzaam is, in GFA een administratiepakket ontworpen buiten zijn overvolle weektaak. Voor leerkrachten, die het pakket in school willen gebruiken, wordt gevraagd een bijdrage van f 50,- te storten op de rekening van Henk. Uiteraard is het pakket pasklaar gemaakt voor de basisschool in Ridderkerk, maar voor eventuele aanpassingen voor een school in Hoogeveen is Henk niet te beroerd. Het pakket werkt in kleur en monochroom. Wat meteen opvalt, is dat ook dit pakket er wat MS-DOS-achtig uitziet. Nogal wat functies dienen met het toetsenbord te worden uitgevoerd. Grafisch niet erg sterk, maar wel functioneel is het persoonsregister georganiseerd. Het afdrukken van nogal wat lijsten is zonder meer mogelijk.
Het aardige aan dit pakket is zeker een apart administratieprogramma, waarmee in beperkte mate boekhouding mogelijk is. In een oogopslag is te zien, wat de inkomsten en de uitgaven van de school zijn. Afdrukken van uit dit onderdeel is natuurlijk ook mogelijk. Aan dit pakket is goed te zien, dat een onderwijsfiguur bezig is geweest. Onder de deskbalk is te lezen, dat het programma niet kompleet is. Het is dus niet af. Daar staat tegenover, dat men het eerst flink kan uitproberen en als het later bevalt, kan men alsnog die f 50,- betalen. Voor inlichtingen: tel. 01804-26731

## BASLAD

In Voorburg zit de firma BITFIT, die administratieprogramma's voor MS- DOS, de Mac, de Amiga en sinds kort ook voor de Atari ST maakt. Zeer waarschijnlijk schrijft de programmeur in een universele taal, die zonder veel poespas over te zetten is naar andere systemen. BASLAD staat voor BASisschool Leerlingen ADministratie. In dit pakket is de muis geheel overbodig. De pijltjestoetsen en de RETURN-toets hebben de handige muis weggedrukt. De grafische layout van elk scherm in BASLAD doet elke gebruiker meteen denken, een oude IBM voor zich te hebben met een zwart-wit gekleurd amber scherm. Het lijkt, alsof dit pakket zonder sleutelen overgezet is naar de ATARI ST. Het ziet er vreselijk belabberd uit. Kopen kan men dit pakket niet. Het pakket wordt namelijk voor f 200,- per jaar verhuurd. Daar staat dan tegenover, dat de programmeur door blijft ontwikkelen en dat de updates in de huurprijs zitten.
Zoals we reeds opmerkten, nodigt het scherm totaal niet uit om in het programma eens rond te neuzen. Op een vreselijk primitieve wijze staan wat ASCII-achtige figuurtjes gegroepeerd en daarbinnen is dat pakket dan opgemaakt. Het pakket wordt op twee schijven geleverd en met een losbladige stencilachtige handleiding. Nadat een TTP-applikatie aangeklikt is, moet er een toegangscode ingetikt worden. Later volgt dan het tweede toegangsnummer en dan verschijnt het hoofdmenu waar, U raadt het al, niet alle opties te gebruiken zijn.

Eerst een jaar huur betalen, want het hoofdmenu is niet geheel gratis. We zien het al voor ons. We leasen een splinternieuwe auto in de garage en als we een jaar lang steeds netjes betaald hebben, krijgen we de verlichting. Oh, wat zijn we dan blij, want nu kunnen we ook 's avonds eens weg. BASLAD kent in de definitieve versie een Leerlingenbestand, Correspondentie, Tellingen, Ouderbijdragen, Personeelsbestand, een Budget en een Tekstverwerker. Nogmaals, alles aan het werk ziet u dus niet. Aan dit pakket zit een voordeel: het werkt zonder PC-DITTO. Voor inlichtingen: tel. 070-990979

## NUMERUS

Sjouke Hamstra komt ook uit het MAVO-onderwijs. Als ex-wiskunde-leraar is Sjouke in zijn vrije tijd gaan programmeren in GFA-Basic versie 3. Helaas is de compiler nog niet uit, hetgeen voorlogig inhoudt, dat eerst een run-only PRG moet worden opgestart. Het vijfde pakket in deze rubriek heet NUMERUS. Eigenlijk is hier sprake van een mini-spreadsheet. Alle mogelijke punten en scores worden door de leerkracht ingevoerd en opgeslagen. Het programma verwerkt deze notaties niet alleen, maar berekent steeds de gemiddelden. Grafisch weergeven van de scores is natuurlijk ook mogelijk. Het programma berekent niet alleen gemiddelden uit, maar ook standaard deviaties en varianties. Het pakket is te koop voor f 69,-, wordt geleverd op een schijf met handleiding. Voor inlichtingen: tel. 033-940611

## Type-kursus

De Stichting GFA heeft weer eens een pakket op de markt gebracht. Het is een typekursus. Als het programma wordt opgestart, komen zeer bekende oefeningen voor, die alle met een test worden afgesloten. Als de gebruiker niet weet op welke toets de vingers gezet moeten worden, dan kan deze dat middels een optie opvragen. Helaas wordt dit onderdeel niet apart ingeoefend, waardoor het blind leren typen niet aan de orde komt. Wat wel kan met dit program, is de typevaardigheid opvoeren. Als men met tien vingers kan typen, is het zonder veel moeite mogelijk om een snelheid te halen van 270 aanslagen per minuut. Daar is menige Schoeffers secretaresse knap jaloers op. Maar typt U helaas met maar twee vingers, dan kunnen die acht na de kursus toch niet echt veel meer. Voor het opvoeren van de typevaardigheid en vooral de snelheid is het programma voor f 45,- niet te veel geld. Voor inlichtingen: tel. 023- 336946

## Rekenen

De Stichting GFA is bezig alle rekenprogramma's grondig te bekijken en daar waar nodig is, deze te onderwerpen aan een grondige renovatie. De bedoeling is, om alle uitgebreide pakketten die vroeger op meerdere schijven geleverd werden, nu m.b.v. moderne technieken terug te brengen naar een schijf. Ook wat prijs betreft, streeft men ernaar alle pakketten op één lijn te zetten. Het zal er zeer waarschijnlijk naar gaan uitzien, dat alle pakketten per stuk f 45,- gaan kosten. Sinds kort is men bij de Stichting GFA bezig een cursus op te zetten voor beginners en gevorderen in GFA-Basic. Voor inlichtingen: tel. 023-336946

## Diktee Extra

Compudress Educatief heeft op het programma DIKTEE een vervolg uitgebracht, DIKTEE EXTRA. Dat EXTRA is inderdaad nogal een toevoeging. Het programma werkt helaas alleen in de lage resolutie. Nou staan er ook zeer fraaie plaatjes in, die op zwart-wit er niet zouden uitzien, maar dan nog. In een tijd, dat menige programmeur ernaar streeft alle systemen van de Atari tegelijk aan te spreken, moet dat toch maar kunnen.
Als het INTRO-programma is aangeklikt, komt men in een zeer apart fraai en rustgevend scherm terecht van alles wat maar op groen lijkt. Hierin kan men kiezen uit dat stuk EXTRA, namelijk uit vijf verschillende uitvoerige spellen. Verder is het ook mogelijk om een hele rits oefeningen te laten uitprinten. Zomaar een taalboek vol met oefeningen rolt dan uit de printer. Keurig. Wat de soorten oefeningen betreft is men niet zuinig geweest. Letterblokken moeten worden aangeklikt, om daar een woord in te ontdekken. Woorden worden in bijna spiegelbeeld aangeboden en de leerling moet dit dan recht zetten. Woorden, die voor een klein deel verschijnen, moeten worden afgemaakt. Dan kan men woorden raden. Gelukkig is hier niet gebruik gemaakt van een galg, maar van een Eskimo die langzaam maar zeker van een ijsberg afglijdt en alsnog een nat pak krijgt. Met dit programma krijgt men voor ongeveer f 60,- een zee aan oefeningen om bij de leerlingen het woordbeeld in te slijpen. Voor inlichtingen: tel. 03481-2124

Jos Smeets

# MIDI IN - MIDI OUT

**In deze aflevering van de Midi In - Midi Out rubriek wordt verder ingegaan op het programmeren van Midi vanuit (GFA-)Basic. De interne afhandeling en verwerking van binnenkomende Midi- informatie wordt toegelicht aan de hand van enkele voorbeelden en Midi-routines. Centraal hierbij staat de Midi-buffer in de ST. Aan het einde van de rubriek volgen nog enkele mededelingen, onder andere over de nieuwe software van Steinberg Research.**

## Midi-buffer

Voor het ontvangen en opslaan van binnenkomende Midi-informatie heeft de ST een Midi-buffer. Midi-bytes die bij de Midi-In-poort van de ST aankomen worden 'automatisch' in deze buffer gezet. Dit gebeurt automatisch, omdat het Operating System van de ST (TOS) hiervoor zorgt (zodra de machine wordt aangezet). De buffer bestaat uit een stuk geheugen waar de Midi-bytes in worden opgeslagen. Bij deze buffer horen weer een aantal gegevens, die onder meer aangeven waar de buffer zich in het geheugen bevindt (m.a.w. het beginadres), hoe groot de buffer is, en hoeveel bytes er reeds in staan. We zullen laten zien hoe deze informatie (via 'pointers') in een (GFA-)Basic-programma is op te vragen en eventueel te veranderen. Hiermee is het mogelijk om bijvoorbeeld zelf een grotere Midi-buffer te installeren in de ST. Dit is onder meer nodig om grote hoeveelheden Midi-data in één keer te kunnen ontvangen, bijvoorbeeld sound-dumps van een synthesizer. Een andere mogelijkheid die gedemonstreerd wordt, is een (software-matige) Midi-Thru.

De Midi-In-buffer werkt volgens het FIFO-principe (First In First Out). Dit houdt in dat de Midi-bytes in [illegible] volgorde van binnenkomst uit de buffer w[illegible]lezen. Dit lezen van Midi-bytes uit [illegible] kan in GFA-Basic onder andere m[illegible]p(3)'. (zie de Midi In - Midi Out-rubriek in ST18). Het gebruik van Inp(3) en andere routines in GFA en de gevolgen die dit heeft voor de Midi-buffer zullen in dit artikel aan de hand van enige voorbeelden worden toegelicht. Een belangrijk punt in dit verband is dat het in principe niet mogelijk is om één byte meerdere keren uit de buffer te lezen. Dit heeft tot consequentie dat er voor de buffer een organisatie moet zijn die bijhoudt waar de eerste en de laatste byte staan. Komt er een byte binnen, dan dient deze achter de laatst binnengekomen byte gezet worden; wordt er een byte gelezen, dan is na deze leesaktie de daaropvolgende byte de eerste geworden.

In GFA kunnen de gegevens die betrekking hebben op de Midi- buffer worden opgevraagd middels een Xbios-aanroep (TOS-routine). Deze aanroep levert het adres waar deze gegevens over de buffer in het geheugen staan. Hierbij gaat het om de volgende gegevens:

- Adres van de Midi-buffer (Long)
- Grootte van de buffer (Word)
- Buffer-head (Word) - Buffer-tail (Word)

**LISTING 1**

```
Mdb_par%=Xbios(14,2)
'
Mdb_addr%=Lpeek(Mdb_par%)           ! startadres van de buffer
Mdb_size%=Dpeek(Mdb_par%+4)         ! buffer-grootte
Mdb_head%=Dpeek(Mdb_par%+6)         ! buffer-head
Mdb_tail%=Dpeek(Mdb_par%+8)         ! buffer-tail
```

Behalve deze zijn er ook nog de Low-mark en de High-mark, maar deze worden bij de Midi-buffer niet gebruikt.
In listing 1 kunt u zien hoe het opvragen van gegevens over de buffer in z'n werk gaat.

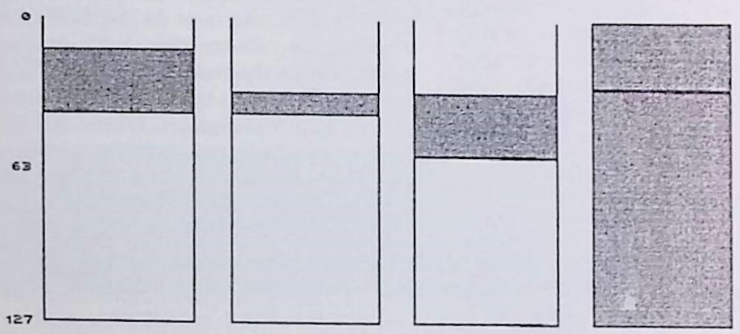
Na de Xbios aanroep staat in de variabele Mdb-par% het adres waarvanaf we de gegevens kunnen vinden (De '14' in de aanroep geeft het Xbios-functienummer weer, de '2' geeft aan dat het om de Midi-buffer gaat, en niet de RS-232 buffer bijvoorbeeld). Hierna worden de relevante gegevens aan de hand hiervan uit het geheugen gelezen.

## Buffer-head & tail

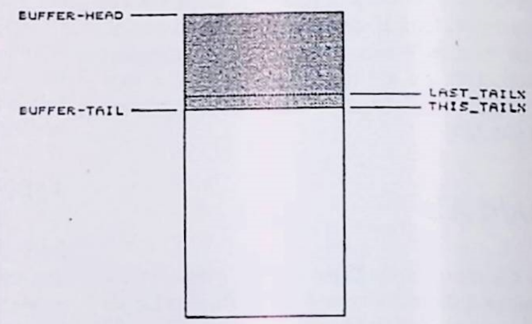
Om een verklaring te geven van de betekenis en verwerking van de buffer-head en de buffer-tail, dienen we ons te verdiepen in de werking van de buffer. Al eerder is vermeld dat elke byte die binnenkomt steeds als laatste gezien wordt ten opzichte van bytes die al eerder zijn binnengekomen. Een nieuwe byte wordt dan ook vooraan in de buffer gezet. Elke ('oude') byte die aan de achterkant uit de buffer wordt gelezen met INP(3) zorgt ervoor dat de byte die er daarachter staat in de buffer nu de volgende eerste wordt.

Als we kijken in figuur 1, dan zien we daar vier afbeeldingen van de Midi-buffer. In de eerste (meest linkse) situatie staan er een aantal bytes te wachten. We zien dat de plaatsen 15 t/m 40 in de buffer bezet zijn. In deze situatie geeft de

*Figuur 1.*




*Figuur 3.*

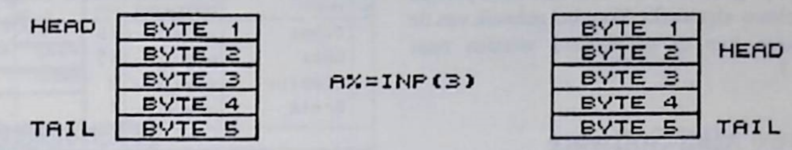


buffer-head 40 aan en de buffer-tail dus 15. In de tweede situatie vinden we de buffer terug nadat er een 15 bytes uit de buffer gelezen zijn. De buffer-head en tail geven nu respectievelijk 40 en 30 aan. In de derde situatie is er Midi-informatie binnengekomen, zodat deze er in de buffer is bijgekomen. De buffer-head en tail zijn nu resp. 60 en 30. Zo zien we dus dat de informatie in de buffer altijd op dezelfde plaats bijft staan en dat alleen de pointers (head en tail) worden gebruikt om aan te geven welk deel van de buffer op dat moment in gebruik is. Zou de buffer nu aan z'n top komen (in de afbeelding aan de onder-zijde), dan wordt de daaropvolgende informatie weer aan het begin van de buffer gezet. Dit houdt uiteraard op wanneer de buffer werkelijk vol is, zoals in de vierde situatie is weergegeven. De buffer-head en tail staan hierbij op resp. 29 en 30. Alle informatie die er nu binnenkomt wordt wel in de buffer geplaatst, maar zal daardoor oude Midi- bytes die nog niet zijn gelezen overschrijven, waardoor de oude informatie verloren gaat.

Toen ik een tijdje geleden werkte aan een sound-editor/memory bank voor mijn electronische drums, stuitte ik op een probleem. Indien ik mijn electronische drums vroeg alle parameters van de instrumenten naar mijn ST te zenden, leek dit steeds niet goed te gaan; de checksum klopte niet aan het einde van het overzenden. Toen ik er uiteindelijk achter kwam dat mijn drums in één keer 134 bytes overzonden, begon mij een lampje te branden. Immers de Midi-buffer van de ST heeft een standaard grootte van 128 bytes. Na die 128 bytes binnen te hebben gekregen overschreef de ST de eerstvolgende 6 bytes in de buffer. En die 6 bytes waren juist de eerste 6 bytes van de dump. Met andere woorden, er kunnen nooit meer Midi-bytes in de buffer worden geplaatst dan er ruimte voor is. Er restte slechts één oplossing: de buffer vergroten.

De routine in listing 2 biedt de mogelijkheid om een grotere Midi-buffer in de ST te plaatsen. Deze buffer moet op een andere plaats in het ST-geheugen worden geplaatst dan de originele buffer. Toch merkt de ST (TOS) hier zelf niets van, en zet keurig de binnenkomende bytes in de nieuwe buffer. De routine werkt als volgt. Allereerst wordt het adres opgevraagd, waar de pointers van de buffer kunnen worden gevonden. Daarna worden de oorspronkelijke pointers bewaard. Deze dienen altijd aan het eind van een programma teruggezet te worden, om het systeem niet te verstoren of eventueel zelfs te laten crashen. Vervolgens wordt er een tabel gedimensioneerd. De voor deze tabel gereserveerde geheugenruimte wordt daarna als buffer gebruikt. Hiervoor wordt om te beginnen het startadres van de tabel opgevraagd en als start van de buffer 'weggepoked'. Vervolgens worden de buffergrootte, de head, de tail (en voor de zekerheid de low- en high-mark) weggeschreven. Het resultaat: een uitstekend functionerende Midi-buffer van 1 Kb.

*Figuur 2.*




## Midi Thru

Zoals enkele van u mogelijk al ervaren hebben, levert het maken van een set-up met meer dan twee instrumenten (en een ST) al vaak problemen op. De instrumenten kunnen immers op verschillende wijzen op elkaar worden aangesloten en in het gunstigste geval werkt er slechts één van deze set-ups. Op de DDR-30 van Roland zit een Midi-Mix optie die ervoor zorgt dat ook het Thru-signaal op de Out-poort wordt gezet. Dit is een handige optie, die echter door weinig instrumenten wordt ondersteund. Zo'n Midi-Mix kan een ieder echter veel problemen besparen en dus zou het prettig zijn wanneer u in staat bent om de ST binnen uw eigen programmatuur deze functie te laten ondersteunen. Eenvoudig gezegd komt het erop neer dat iedere byte die binnenkomt er direct ook via de Out-poort weer uit gaat. De alom bekende Midi-Thru dus, maar dan gecombineerd met de Out.

Hierbij stuiten we echter op een probleem. Zouden we bijvoorbeeld de volgende regels aan ons programma toevoegen

```
If Inp?(3)
Out 3,Inp(3)
Endif
```

dan verliezen we hierbij wel alle Midi-informatie. Zoals ik al eerder aangaf, kunnen we elke byte slechts één keer uit de buffer lezen met Inp(3). Het probleem ligt hierbij in de administratie van de buffer-head. Deze wordt namelijk na elke leesaktie één positie naar voren verschoven. Dit wordt gedemonstreerd in figuur 2. De bytes blijven gewoon in de buffer staan, maar aangezien de Inp(3) functie de byte teruggeeft waar de buffer-head naar wijst en deze daarna automatisch ophoogt, krijgt u elke byte slechts één keer 'te zien'.

In het vorige voorbeeld hebben we echter gezien hoe we de stand van de buffer-head kunnen opvragen. Al wat we dan nog hoeven te doen is steeds de oude stand te onthouden en rechtstreeks de bytes die na de laatste keer zijn binnengekomen uit de buffer te lezen. Dit rechtstreeks lezen van de buffer kan met de peek-operstie in Basic. Deze verandert namelijk niet de buffer-tail. Op de in listing 3 gedemonstreerde methode beschikken we binnen onze applicatie nog via de normale Inp(3) aanroep over de Midi- informatie die op onze ST binnenkomt. Het is een voorbeeld van hoe de Midi-buffer en de buffer-gegevens met peek- en poke- operaties kunnen worden bewerkt.

In het begin van het programma wordt er allereerst een aanroep gedaan aan de procedure Empty-buf. Deze leegt de Midi-buffer en zet de head en tail beide op nul (we beginnen met een schone lei). Daarna komt het programma in een lus die slechts door het toetsen van Escape kan worden onderbroken. Aan het begin van de lus wordt de huidige tail bepaald (This-tail%). Daarna wordt deze met de head vergeleken om te kontroleren of de buffer niet vol is. Is de buffer vol, dan kan er worden gekozen tussen het beëindigen van het programma of het legen van de buffer en het daarna voortzetten van de lus. Hierna wordt de huidige tail met de 'laatst gemeten' (Last-tail%) vergeleken en worden de bytes die er kennelijk in de tussentijd zijn

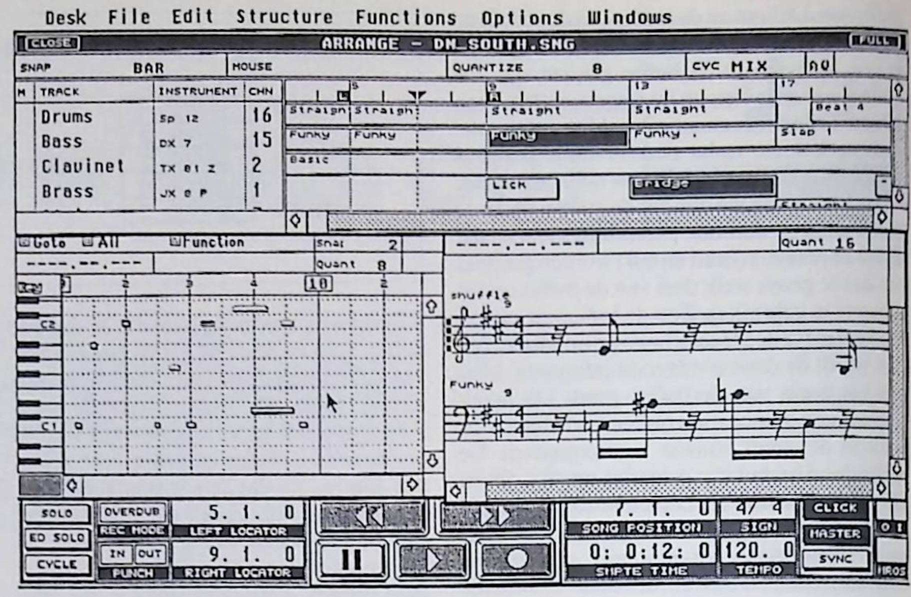
**LISTING 2**

```
' ***********************************************************
'                   INCREASE MIDI-BUFFERSIZE
'
'                      Vincent van Diemen
' ***********************************************************
'
Mdb_par%=Xbios(14,2)                     ! Adres van de pointers
'
Old_strt%=Lpeek(Mdb_par%)                ! Oorspronkelijk startadres
Old_size%=Dpeek(Mdb_par%+4)              !    "      "     grootte
Old_head%=Dpeek(Mdb_par%+6)              !    "      "     buffer-head
Old_tail%=Dpeek(Mdb_par%+8)              !    "      "     buffer-tail
Old_lmrk%=Dpeek(Mdb_par%+10)             !    "      "     Low mark
Old_hmrk%=Dpeek(Mdb_par%+12)             !    "      "     High mark
'
Dim Mdb%(256)                            ! Tabel van 1024 bytes (*)
'
Lpoke(Mdb_par%),Varptr(Mdb%(0))          ! Nieuwe startadres
Dpoke(Mdb_par%+4),1024                   !    "    grootte        (*)
Dpoke(Mdb_par%+6),0                      !    "    buffer-head
Dpoke(Mdb_par%+8),0                      !    "    buffer-tail
Dpoke(Mdb_par%+10),0                     !    "    Low mark
Dpoke(Mdb_par%+12),1023                  !    "    High mark
'
'
'    Uiteraard kan de grootte van de buffer naar wens
'    worden ingesteld door de dimensionering van de
'    tabel en het wegpoken aan te passen (*).
```

bijgekomen ge-peek-ed en via de Midi-poort uitgezonden (thru-functie). Tenslotte wordt This-tail% gelijk gemaakt aan Last-tail% en wordt de lus opnieuw afgewerkt. Voor het gebruik van de variabelen kan er gerefereerd worden naar figuur 3.

## Nieuwe Midi-Software

Op de Frankfurter Messe van afgelopen februari stonden weer een groot aantal Atari's met Midi-software. Hoewel er van een aantal programma's wel updates te zien waren, was er over het algemeen weinig echt nieuws te melden. De interessantste aankondigingen en demonstraties kwamen van Steinberg. Naast een uitgebreide editor voor de Korg M-1 'music workstation' werd ook een editor voor de Casio FZ-1 sampler geintroduceerd. Avolon is een uitgebreide sample-editor die op verschillende synthesizers werkt. Het meest interessant waren echter Steinberg's nieuwe versies van het sequencer pakket Pro-24 III. Het programma kent zelf een nieuwe update en heeft daarnaast ook een kleiner broertje gekregen in de vorm van 'Twelve'. Dit is een 12-sporige midi-sequencer met ongeveer dezelfde functies, maar eenvoudiger van opzet. Het programma is duidelijk gericht op de beginnende Midi-gebruiker. Er komt echter binnenkort ook een professionele sequencer op de markt onder de naam Cubit. Dit programma is duidelijk een krachtigere uitvoering van Pro-24. Het meest opvallend is de consistente window-interface: alle functies hebben nu een eigen window (zie figuur 4). Bovendien zijn er weer een heel wat mogelijkheden bijgekomen. Het programma is gebaseerd op een nieuw Operating System voor de ST, genaamd M-ROS. Dit besturingssysteem is speciaal voor real-time Midi-applicaties geschreven, en het vervangt de bekende TOS. Het maakt het zelfs mogelijk om verschillende Midi-applicaties real-time parallel te laten draaien. Op deze nieuwe ontwikkelingen rond Cubit en M-ROS zullen we in het volgende nummer van ST terugkomen.




*Figuur 4: De nieuwe professionele sequencer Cubit van Steinberg.*

**LISTING 3**

```
' ***************************************************************
'
'                    NON-DESTRUCTIVE MIDI-THRU
'
'                        Vincent van Diemen
' ***************************************************************
'
Box 245,30,370,49
Print At(32,3);Chr$(27);"p   Midi-Thru   ";Chr$(27);"g"
'
@Empty_buf
'
Mdb_size%=128
'
Do
  '
  This_tail%=Dpeek(Xbios(14,2)+8)              ! Current tail
  '
  If This_tail%=Dpeek(Xbios(14,2)+6)+Mdb_size%-1
    Alert 3,"|Midi buffer full!",1," Clear | Stop ",Cos%
    If Cos%=1
      @Empty_buf
    Else
      End
    Endif
  Endif
  '
  If This_tail%<>Last_tail%                    ! Any new bytes?
    '
    If This_tail%<Last_tail%                   ! Past the top?
      Add This_tail%,Mdb_size%
    Endif
    '
    For Thru%=Last_tail%+1 To This_tail%
      Out 3,Peek(Lpeek(Xbios(14,2))+Thru% Mod Mdb_size%)
    Next Thru%
    '
    Last_tail%=This_tail% Mod Mdb_size%        ! Equal both
    '
  Endif
  '
  Exit If Asc(Inkey$)=27                       ! Exit if Esc
  '
Loop
'
Procedure Empty_buf
  '
  While Inp?(3)                                ! Clear buffer
    Void Inp(3)
  Wend
  '
  Dpoke (Xbios(14,2)+6),0                      ! Initial head
  Dpoke (Xbios(14,2)+8),0                      ! Initial tail
  '
Return
```

Het sequencer-programma Master Tracks Pro van Passport Designs heeft eindelijk een Nederlandse importeur gekregen. Amazing Enterprises in Den Haag zal de software van Passport Designs gaan voeren. Naast de Pro-versie van Master Tracks is er nu ook een Jr uitvoering voor de beginnende gebruikers.

## Landelijke Midi Dag

Op zaterdag 18 maart heeft de Dutch Midi Club (DMC) een landelijke Midi-dag georganiseerd in Hazerswoude. Er waren een aantal Nederlandse importeurs van Midi-software aanwezig, en daarnaast werden er een aantal productdemonstraties verzorgd door Roland. Ook waren er enkele computergebruikers met Midi-demonstraties. Over de gehele dag zijn er zo'n 150 mensen langsgekomen die deze eerste landelijke Midi-dag in het algemeen positief waardeerden.

**Vincent van Diemen en Guus Ramackers**

# Logo Programmeurcursus

## voor kinderen (en volwassenen) aflevering 7

**Beste jongens en meisjes, vaders en moeders!**
**De definitieve Nederlandstalige versie van LOGO voor de ATARI-ST kun je samen met de HANDLEIDING bestellen tegen kostprijs bij het LOGO CENTRUM EDE. Het adres (ook voor hele aardige LOGO-werkboekjes) is:**

**Logo Centrum Ede, Annadaal 96, 6715 JC Ede. Tel.: 08380 - 21306**

### Inleiding

De vorige aflevering (ST-BLAD nummer 18 van maart/april 1989) hebben we besteed aan de simulatie van een simpele auto. Déze aflevering willen we besteden aan de simulatie van een betere auto en het begrip 'trage massa'. De snelheid van deze betere auto kan dan worden verhoogd en verlaagd. Het doel van deze simulatie is, dat je wat meer leert over de besturing van een auto bij wat hogere snelheden. Bovendien komen de begrippen snelheid, snelheidsverandering (acceleratie of versnelling), trage massa, wrijving (in de vorm van luchtweerstand en rolweerstand), stroomlijn (Cw-waarde) en remweg om de hoek kijken!

### Logo opstarten en bestand intypen

Voordat de computer-apparatuur wordt aangezet, moet je zorgen dat de LOGO- schijf zich in de diskdrive bevindt. Als de apparatuur is ingeschakeld, zie je op het beeldscherm de bestanden staan. Start het LOGO.PRG op.

Als de LOGO-vensters op het scherm zijn verschenen met de menubalk erboven, typ je het volgende LOGO-bestand in (sluit iedere regel af met <RETURN>):

```
LEER AUTO
VB VEEGUIT ROND
MAAK "SNELHEID 0
(DRUK [SNELHEID IS NU] :SNELHEID
[KM PER UUR])
RIJDEN
EIND

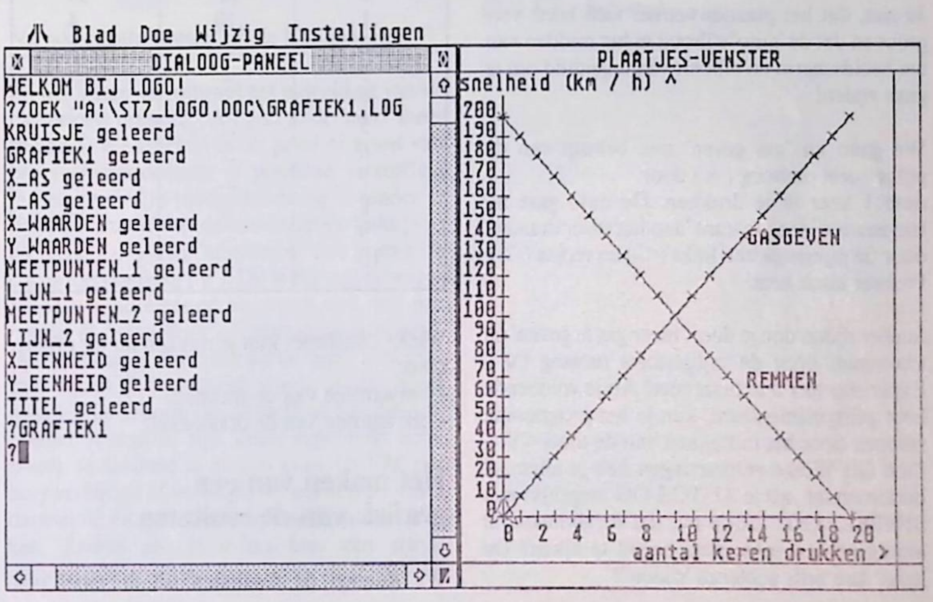
LEER RIJDEN
ALS TOETS? [LEESTOETS]
VOORUIT :SNELHEID / 10
RIJDEN
EIND

LEER LEESTOETS
MAAK "TOETS LEESLETTER
ALS :TOETS = DRUKLETTER 6 [RECHTS
10] ;ascii-code cursorpijltje rechts
ALS :TOETS = DRUKLETTER 2 [LINKS
10] ;ascii-code cursorpijltje links
ALS :TOETS = DRUKLETTER 16 [VER-
SNEL 10] ;ascii-code cursorpijltje omhoog
ALS :TOETS = DRUKLETTER 14 [VER-
SNEL -10] ;ascii-code cursorpijltje omlaag
ALS :TOETS = "S [DOENU "TOPNIVO]
EIND

LEER VERSNEL :VERANDERING
MAAK "SNELHEID :SNELHEID + :VER-
ANDERING
(DRUK [SNELHEID IS NU] :SNELHEID
[KM PER UUR])
EIND
```

Bewaar het ingetypte bestand op een geformatteerde, niet schrijfbeveiligde diskette door in te typen:

BEWAAR "AUTO.LOG <RETURN>



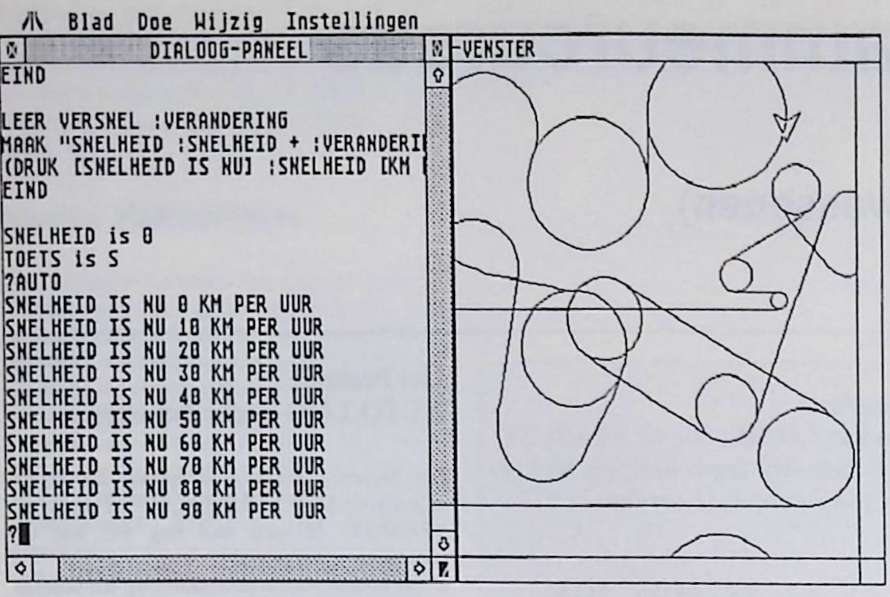

### Het bestand AUTO.LOG nader bekeken

Het bestand AUTO.LOG bestaat dus uit 4 procedures: AUTO, RIJDEN, LEESTOETS en VERSNEL. Je weet vast nog wel wat de commando's waren om:
a. De namen van de procedures op het scherm te krijgen (DP of DPN);
b. De definities van de procedures op te vragen (DAALLES of DA "procedurenaam). Probeer beide maar eens uit! Eventueel kun je het dialoog-paneel van te voren nog vergroten door te klikken op de 'full-box' in de rechter bovenhoek van het dialoog-paneel. Nogmaals klikken na afloop verkleint het dialoog-paneel weer tot de oorspronkelijke grootte.

Als je een printer hebt aangesloten, kun je de bestanden van AUTO.LOG proberen uit te printen. Hiertoe typ je het volgende in (zet de printer van te voren aan met papier erin):

?PRINTER
?DAALLES
?PRINTERUIT

Probeer nu de volgende vragen te beantwoorden:
- Wat is de naam van de variabele die in de procedure VERSNEL is opgenomen?
- Welke procedure van de vier hierboven is recursief (roept zichzelf aan)?
- Welke procedure wordt na het starten slechts 1 keer doorlopen en verder niet meer aangeroepen?




- Welk speciale teken wordt in LOGO gebruikt, om daar achter opmerkingen of commentaar in een LOGO-procedure te vermelden, zonder dat dit door de LOGO-taal wordt geïnterpreteerd en uitgevoerd?
- Welke procedure is toegevoegd in vergelijking met SIMPEL en SIMPELAUTO uit de vorige aflevering (ST-BLAD 18, maart/april '89)?
- Heb je ook enig idee waarom er een procedure is toegevoegd en waarom sommige andere procedures zijn veranderd en uitgebreid?
- Welke verbeteringen denk jij dat er zijn aangebracht in vergelijking met SIMPEL? Bekijk hiervoor de lijst met procedures van AUTO.LOG en vergelijk die met de procedures van SIMPEL.LOG!

## Het uitvoeren van de auto-simulatie

Om het bestand AUTO.LOG te testen en om te kijken, of de 'auto' ook naar behoren functioneert, starten we het programma op door intypen van:

?AUTO <RETURN>

Je ziet, dat het plaatjes-venster zich heeft vergroot en dat de 'auto' stilstaat in het midden van het beeldscherm en bibbert van ongeduld om te gaan rijden!

We gaan nu 'gas geven' met behulp van de pijltjestoets omhoog (∧) door
deze 1 keer in te drukken. De auto gaat nu langzaam rijden. Je kunt 'aan het stuur draaien' door de pijltjestoetsen links (<-) en rechts (->). Probeer maar eens!

Sneller rijden doe je door 'meer gas te geven' en afremmen door de pijltjestoets omlaag (∨). Experimenteer hier maar mee! Als je voldoende hebt geëxperimenteerd, kun je het programma stoppen door het indrukken van de toets <S>. Doe dit! Welke verbeteringen heb je allemaal ondervonden, als je AUTO.LOG vergelijkt met SIMPEL.LOG? Zie je nu, dat de simulatie al aardig op een echte auto begint te lijken? De 'auto' kan zelfs achteruit 'rijden'!!

## Opdracht

We gaan nu wat metingen aan het gedrag van onze 'auto' verrichten. We gaan de snelheden noteren van de 'auto' bij diverse keren 'gas geven' en meten met een geo-driehoek of lineaal op, hoe groot de diameter is van de kleinste cirkel die we bij iedere snelheid kunnen maken (door constant in één bepaalde richting te sturen). Enkele vakjes zijn in de tabel al ingevuld. Aan jou de taak om de andere vakjes in te vullen, terwijl je met de simulatie bezig bent! Zorg, dat je opnieuw begint (eventueel <S> indrukken). Start weer door het intypen van AUTO <RETURN> en vul de tabel verder in tijdens het experiment. Steeds één keer tegelijk op pijltje omhoog drukken!

| Aantal keren gas geven (∧): | Snelheid in KM per uur: | Diameter kleinste draaicirkel (in mm): |
|---|---|---|
| 0 | ..0 | ..0 |
| 1 | .10 | ..4 |
| 2 | ... | ... |
| 3 | ... | ... |
| 4 | ... | ... |
| 5 | ... | ... |
| 6 | ... | ... |
| 7 | ... | ... |
| 8 | ... | ... |
| 9 | ... | ... |
| 10 | ... | ... |

Welke conclusies kun je trekken uit dit experiment:
- ten aanzien van de snelheid?
- ten aanzien van de draaicirkel?

## Het maken van een grafiek van de rsultaten

We kunnen de resultaten uit de tabel ook overzichtelijk in een grafiek weergeven. Om deze grafiek te voorschijn te toveren, moet je het programma AUTO eerst stoppen met de toets <S> en daarna het onderstaande intypen (steeds iedere regel afsluiten met <RETURN>):

```
LEER GRAFIEK1
X-AS
Y-AS
X-WAARDEN
Y-WAARDEN
X-EENHEID
Y-EENHEID
MEETPUNTEN-1
LIJN-1
MEETPUNTEN-2
LIJN-2
TITEL
EIND

LEER X-AS
PENOP
ZETPOS [-120 -140]
RICHTOP 90
PENNEER
VOORUIT 255
ACHTERUIT 255
HERHAAL 21 [LINKS 90 VOORUIT 4 ACHTERUIT 8 VOORUIT 4 RECHTS 90 VOORUIT 12]
PENOP
RICHTOP 0
ZETX -120
EIND

LEER Y-AS
PENOP
ZETPOS [-120 -140]
RICHTOP 0
PENNEER
VOORUIT 295
ACHTERUIT 295
RECHTS 90
ACHTERUIT 4
VOORUIT 4
LINKS 90
HERHAAL 20 [VOORUIT 14 RECHTS 90 VOORUIT 4 ACHTERUIT 8 VOORUIT 4 LINKS 90]
PENOP
ZETY -140
EIND

LEER X-WAARDEN
PENOP
ZETPOS [-120 -140]
RICHTOP 0
ACHTERUIT 17
RECHTS 90
MAAK "X 0
HERHAAL 11 [PENNEER SCHILDPADTEKST :X MAAK "X :X+ 2 PENOP VOORUIT 24]
ZETPOS [-120 -140]
RICHTOP 0
EIND

LEER Y-WAARDEN
PENOP
ZETPOS [-120 -140]
RICHTOP 90
ACHTERUIT 12
```

```
PENNEER
SCHILDPADTEKST 0
PENOP
ACHTERUIT 8
LINKS 90
MAAK "Y 10
HERHAAL 9 [VOORUIT 14 PENNEER
SCHILDPADTEKST :Y MAAK "Y :Y + 10
PENOP]
RECHTS 90
ACHTERUIT 8
LINKS 90
HERHAAL 11 [VOORUIT 14 PENNEER
SCHILDPADTEKST :Y MAAK "Y :Y + 10
PENOP]
ZETPOS [-120 -140]
EIND

LEER X-EENHEID
PENOP
ZETPOS [-30 -170]
PENNEER
SCHILDPADTEKST [aantal keren drukken
>]
PENOP
ZETPOS [-120 -140]
RICHTOP 0
EIND

LEER Y-EENHEID
PENOP
ZETPOS [-150 159]
PENNEER
SCHILDPADTEKST [snelheid (km / h) ∧ ]
PENOP
ZETPOS [-120 -140]
RICHTOP 0
EIND

LEER KRUISJE
RICHTOP 45
PENNEER
VOORUIT 4
ACHTERUIT 8
VOORUIT 4
LINKS 90
VOORUIT 4
ACHTERUIT 8
VOORUIT 4
PENOP
RICHTOP 0
EIND

LEER MEETPUNTEN-1
PENOP
MAAK "X -120
MAAK "Y -140
HERHAAL 21 [ZETPOS LIJST :X :Y MAAK
"X :X + 12 MAAK "Y :Y + 14 KRUISJE]
ZETPOS [-120 -140]
EIND

LEER LIJN-1
PENOP
ZETPOS [-120 -140]
PENNEER
ZETPOS [120 140]
PENOP
ZETPOS [-120 -140]
RICHTOP 0
EIND

LEER MEETPUNTEN-2
PENOP
MAAK "X -120
MAAK "Y 140
HERHAAL 21 [ZETPOS LIJST :X :Y MAAK
"X :X + 12 MAAK "Y :Y - 14 KRUISJE]
ZETPOS [-120 -140]
EIND

LEER LIJN-2
PENOP
ZETPOS [-120 140]
PENNEER
ZETPOS [120 -140]
PENOP
ZETPOS [-120 -140]
RICHTOP 0
EIND

LEER TITEL
PENOP
ZETPOS [50 50]
PENNEER
SCHILDPADTEKST [GASGEVEN]
PENOP
ZETPOS [50 -50]
PENNEER
SCHILDPADTEKST [REMMEN]
PENOP
ZETPOS [-120 -140]
RICHTOP 0
EIND
```

Typ nu in om de grafiek te bekijken:

?VU GRAFIEK 1 <RETURN>

Eventuele fouten nog even verbeteren en daarna 'saven':

?BEWAAR "GRAFIEK1.LOG

Nu enkele vragen naar aanleiding van deze grafiek:
- Wat valt je op?
- Welke tekortkomingen heeft je huidige 'auto' op dit moment nog, als je hem vergelijkt met een echte auto?
- Zou het in principe mogelijk zijn, de simulatie-AUTO nog verder te verfijnen zodat het bijna een echte auto wordt?

Nu nog enkele 'kreten' uit de Natuurkunde:

Bij het 'gasgeven' oftewel het indrukken van de cursortoets omhoog (∧) geef je de 'auto' meer snelheid. We spreken in dit geval ook wel van een snelheidstoename of positieve versnelling (acceleratie). De snelheidsrichting verander je door het 'sturen' met de cursortoetsen links (<-) en rechts (->). Het 'afremmen' doe je met de cursortoets omlaag (∨). Dit is het tegenovergestelde van accelereren en wordt ook wel een negatieve versnelling of vertraging genoemd. Hiermee kun je doorgaan tot de 'auto' achteruit gaat rijden. De snelheid wordt in dat geval negatief! De grootte van de snelheid kan dus nul, positief of negatief zijn. Zoals je gemerkt hebt, wordt de snelheid in stapjes (van 10 KM per uur) verhoogd of verlaagd. Bij een echte auto is dat niet zo: dit gaat continu oftewel ononderbroken. Anders zou je iedere keer een schok voelen!

Je kunt eventueel proberen, de stapgrootte van de snelheidstoename kleiner te maken. Hiervoor moet je het volgende weten:

NIEUWE SNELHEID = OUDE SNELHEID + SNELHEIDSTOENAME

Hierbij merken we nog op, dat de snelheidstoename ook negatief kan zijn; met andere woorden: de snelheid kan dan afnemen!

Als je handig bent en zin hebt, kun je de stapgrootte in de procedure LEESTOETS nog veranderen (deze staat ingesteld op VERSNEL 10 en VERSNEL -10). Wijzigen doe je met WZ "LEESTOETS aan te roepen.

Je hebt gemerkt dat de draaicirkel van de 'auto' bij het nemen van bochten groter werd naarmate de snelheid hoger werd. Bij een echte auto is dat ook zo! Bij het nemen van een bocht wil de 'zware' massa van de auto als het ware rechtdoor. Door wrijving met het wegdek wordt dat grotendeels voorkomen. Echter niet helemaal. De auto gaat dan op z'n banden een stukje doorschuiven. Dat merk je aan het piepen van de banden op het wegdek, iets wat bij hogere snelheden toeneemt en tot slippen aanleiding kan geven! Vooral bij een nat wegdek.
In de Natuurkunde spreken we vaak van het begrip trage massa. De auto is op te vatten als een massa (van ongevéer 1000 kg of 1 ton), die vertraagd (na een poosje pas) reageert op veranderingen in de bewegingsrichting. Een auto reageert dus vertraagd op stuurbewegingen; het effekt hiervan neemt toe bij steeds hogere snelheden! Vandaar, dat in het echt de draaicirkel ook steeds groter wordt (tot de auto in een bocht slipt en ècht uitbreekt!). Het effect van bijsturen wordt dus bij hogere snelheden moeilijker en hiervoor moet je dan ook oppassen! Volledigheidshalve merken we nog op dat deze traagheid van een massa ook optreedt bij accelereren en afremmen. De massa 'schuift' nog een poosje door in z'n oude snelheid!

Veel plezier met LOGO en tot de volgende keer!

**Wouter Schuurmans,**
Archipel 23-39,
8224 GR Lelystad.

# Games

**Deze keer bespreken in de GAME-rubriek drie spellen en een nieuwe medewerker. De spellen zijn, Dragon Scape, Falcon en Lombard RAC Rally. De nieuwe medewerker is Sander Vriesman. Sander zal voor de komende GAME rubrieken een keuze maken uit het aanbod van nieuwe spellen. Hieruit kiest hij dan één spel dat hij uitgebreid zal gaan bespreken. Deze keer is zijn keus gevallen op Lombard RAC Rally.**

## Dragon Scape

*Dragon Scape (kleur)*
*Beschikbaar gesteld door Software Horizons U.K.*
*Nederlandse importeur: HomeSoft Benelux.*

Niets vermoedend zit je te lezen in je favoriete (ST)computerblad. Dan opeens, een harde knal en een oogverblindende lichtflits. Je vind je zelf terug in een vreemde omgeving ver terug in de tijd. Het bleek dat de oudsten van een magisch land, genaamd Tuvania, al hun krachten hebben gebundeld om je door tijd en ruimte te kunnen transporteren om hen te helpen bij een zeer gevaarlijke zaak.
Het land Tuvania leefde alle jaren in vrede. De magische krachten van het land werden al die tijd in evenwicht gehouden door 40 magisch voorwerpen. Maar toen sloeg het kwaad toe. Door een vreemde wending van het lot werd door de boosaardig koning Kaos, die leider van dit alles was, de balans verstoort.

Gelukkig hoef je de boosaardige Kaos niet in je eentje te bestrijden. De oudsten hebben uit alle trouwe bewoners de dapperste krijgers verzameld om je te kunnen helpen. Deze krijgers hebben zich op het grote plein verzameld, nieuwsgierig als ze zijn naar de held die hen van de ondergang komt redden. Dan stapt een van de oudsten naar voren en zegt met een luide stem: „Wie van jullie dappere krijgers wil deze jonge held helpen?" Een doodse stilte daalt neer over het plein. Dan opeens een krakende stem uit de menigte: „Hoe moet hij, die niets van magie weet, ons helpen?" Langzaam beginnen sommige zich om te draaien en weg te lopen van het plein. De oudsten beginnen te wanhopen, totdat een klein stemmetje uit de menigte roept: „Ik wil je wel helpen. Ik mag dan niet zo dapper zijn en heb ook grote magisch krachten hebben, maar ik heb een paar krachtige vleugels." Het is Garvan, een draak. Van alle draken waarover geschreven is, is Garvan misschien wel de stomste draak die je ooit kan tegenkomen. Maar ja, hij is erg aardig en ondanks het protest van de oudsten besluit je om samen met hem ten strijde te trekken tegen de vloek van Kaos.

Nou daar sta je dan. Je moet in het spel vijf verschillende niveaus doorlopen. In elk niveau moet je acht voorwerpen zien te verzamelen, die je op de juiste plaats in het speelveld moet zetten. Om dit geheel te volbrengen zal je diverse wezens moeten vernietigen die je het leven zuur proberen te maken. Ook heeft dit spel een soort cheat mode. Om naar het volgende niveau over te gaan moet je de SHIFT- en de Cursor-downtoets indrukken. Overigens werkt dit maar tot niveau drie.

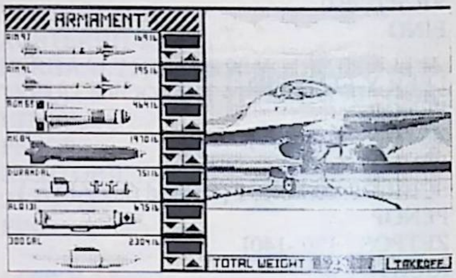
## Falcon

*Falcon (kleur)*
*Richtprijs fl 89,50*
*Beschikbaar gesteld door HomeSoft Benelux.*

Toen ik het spel, nou ja spel, zeg maar F-16 simulator, de eerste keer zag had ik geen woorden om het te beschrijven. En dat is toch wel lastig als je er iets over moet gaan schrijven. Falcon, de F-16 fighter simulatie zoals de verpakking zegt, heeft een zeer verzorgd en snel 3-D grafisch scherm met veel details. Ook is Falcon een flight-simulator waarin je alles kan terug vinden wat er aan, in en op een F-16 te vinden is. Diverse wapensystemen zoals AIM9 en AIM9L heatseeking missiles, AGM-65 Maverick missiles, MK 84 bommen en dergelijke, head-up-display, cockpit controls, kortom alles wat je nodig hebt om een missie met een F-16 tot een goed einde te brengen.




Als je gaat vliegen me deze F-16, kom je van alles tegen op 3-D scherm. Gebouwen, bossen, wegen en bruggen flitsen onder je door als je met Mach-2 op 50 meter hoogte boven de grond vliegt. Ook tanks, vliegvelden en SAM- sites kom je tegen. De vliegtuigen die je tegen komt zijn MiG's en ze lijken er nog op ook, dus niet van die vliegen driehoekjes die je wel eens tegen komt in andere flight-simulators. Afhankelijk van de missie die je gekozen hebt, moet je proberen die bruggen, tanks, vliegtuigen en dergelijke proberen te vernietigen.




De simulator heeft 5 verschillende niveau's waaruit je kan kiezen. Van Luitenant (makkelijk) tot kolonel (moeilijk). In de makkelijkste mode (luitenant) is het niet mogelijk te crashen: je kan van grote hoogte recht omlaag duiken en er gebeurt niets, zelfs als je door een berg heen wil vliegen lukt dat nog. Het prettige van deze mode is dat je bekend kan raken met de vliegeigenschappen van het toestel zonder dat een crash wordt gestraft met een re-start. Anders ligt het als je het niveau van kolonel kiest. Dan is is elke stuurfout fataal, word je aangevallen vanuit de lucht en vanaf de grond en dat met de meest moderne wapens zoals MiGs-21 en SAM-raketten. Dan moet je jezelf verdedigen met alle tot je beschikking staande hulpmiddelen en dat is knap lastig; dan is een crash zeer goed mogelijk. Niet alleen de vliegeigenschappen van de F-16 zijn zeer indrukwekkend, ook de vliegeigenschappen van de tegenpartij zijn indrukwekkend. Als je achter een MiG aan zit dan zal deze alle in de moderne gevechtstechniek bekende trucs gebruiken om je van zich af te schudden. En dat zijn er heel wat.

Falcon wordt geleverd met een 145 pagina's dikke handleiding. Die heb je nodig ook, anders is het heel lastig vliegen. Tenzij je het laagste niveau kiest. Uiteraard staan er de instructies in en hoe je met zo'n F-16 moet vliegen. Ook staan er de diverse missies in beschreven en wat nog veel interesanter is, de diverse vlieg-manoeuvres die je kan uitvoeren als je aangevallen wordt.

De enige min-puntjes die ik bij Falcon kon vinden zijn het geluid en het gecombineerd

vliegen met een andere computer via de RS-232 kabel. Het geluid is een beetje magertjes. Maar gezien de kwaliteit van de rest van het programma valt dit in het niet. Bij het gecombineerd vliegen met een andere computer, de zogenaamde two-player dogfight option, wordt het geheel wat traag. Dit vind ik wel jammer. Wel weer positief is, dat je niet gebonden bent aan een Atari als tweede computer. Je kunt ook kiezen voor een Apple Macintosh of een Amiga als tweede computer, als je daar dan de Falconversie voor hebt natuurlijk.

**Egbert de Rooij**

Het volgende deel is geschreven door Sander Vriesman, onze nieuwe medewerker voor de GAME-rubriek.



## Lombard RAC Ralley

*Lombard RAC Rally (kleur)*
*Richtprijs fl 89,50*
*Beschikbaar gesteld door: HomeSoft Benelux.*

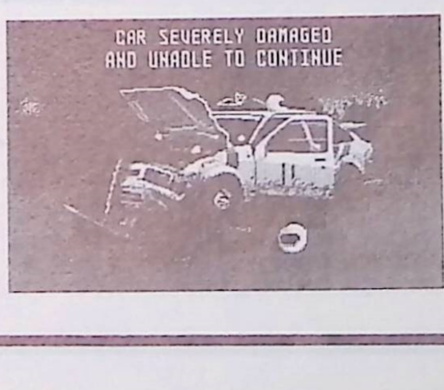
De Engelse handleiding van Lombard Rally bestaat uit 20 bladzijden, verdeeld in 6 verschillende hoofdpunten. Er staat veel geschiedenis in over de rally, hoe het begonnen was, wie de winnaars waren, etc. Er zijn ook 2 stickers bij geleverd van Lombard Rally. Leuk voor op de autoruit. Het spel is alleen te laden in lage resolutie.

Tijdens het laden van het spel hoor je een langsscheurende auto, met daarbij matige digitale muziek. Het laden duurt niet lang. Na het laden wordt er om de DATA-disk gevraagd, die er voor de rest van het spel in de diskdrive moet blijven. In het MAIN-menu aangekomen, kan je kiezen welke routes je gaat rijden; ook kan je eerst even geld gaan verdienen in een TV-interview. Als je een vraag fout beantwoordt, moet je 20 betalen; als het antwoord goed is krijg je 50. De vragen zijn allemaal 3-keuzevragen, dus erg moeilijk is het niet. Als je weinig verstand hebt van rally's moet je veel gokken. Het is niet mogelijk om in schuld te komen, ook niet als je meer verliest dan je hebt; dan is je bedrag gewoon 0. Je kan ook naar de garage gaan en onderdelen van de auto laten maken.

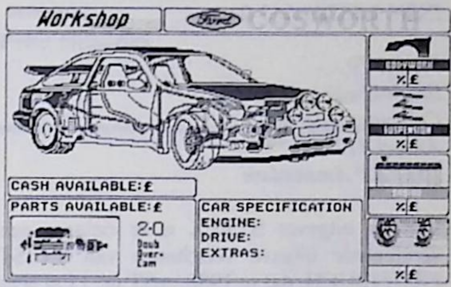



Hiervoor is veel geld nodig. Je kan altijd naar de garage gaan, ook tijdens races, door op 'W' te drukken. Door met de joy-stick te bewegen komt er een blauw vierkant om de onderdelen te staan, Als er genoeg geld is, en er is schade, kan deze schade worden gerepareerd door op de fire-knop te drukken, Als je het ten dele hersteld wilt hebben, druk je twee keer op de knop. Alles kost 1 per procent dat gemaakt wordt. De onderdelen die gemaakt kunnen worden zijn b.v. chassis, vering, motor en de banden. Per crash die gemaakt wordt, komt er steeds meer schade aan de onderdelen; ga je te laat naar de garage, dan is het spel afgelopen. Als je naar de garage gaat tijdens de race, stopt de tijd. Tijdens de race kan je ook pauzeren door op space te drukken. Met ESCAPE ga je direct van de baan af om opnieuw te starten. In MAIN kan je ook voor 'PRACTICE' kiezen; hier zijn zaken uit te proberen, maar hier zijn helaas geen prijzen te winnen. Als je 'DEMO' kiest, zie je hoe er eigenlijk gereden moet worden. Na 3 banen rijden keer je terug naar het hoofdmenu, De DEMO kan je halverwege stoppen met de fire-knop. Als het toetsenbord of joystick 5 minuten lang niet wordt aangeraakt, begint het spel uit verveling vanzelf aan de DEMO.

Als je start, hoor je een stem aftellen, en de race kan beginnen. De bedoeling van het spel is dat je eerst 5 routes rijdt, en ten minste een prijs wint. Zijn deze 5 routes gereden, dan men aan de grote race beginnen. Er kan gereden worden van Harrogate naar Telford of naar Carisle of andersom, of gewoon een rondje in Harrogate zelf rijden. Er zijn moeilijkheidsgraden zoals op mistige-en donkere wegen rijden. Deze moeilijkheidsgraden zijn verdeeld in stages 1, 2 en 3. De routes bestaan uit slingerende berg- en boswegen, De bergwegen zijn slecht gemaakt. Een dikke bruine balk links van het scherm moet een berg voorstellen. En de rotsen lijken op gehaktballen. Alle wegen zijn zeer moeilijk te overzien. Plotselinge bochten en andere auto's zorgen er vaak voor dat je van de weg afschiet, zodat er dan helemaal overnieuw geschakeld moet worden. of, als het een harde klap was, er moet overnieuw gestart worden. Bij een harde botsing zie je wat onherkenbare onderdelen voor de autoruit verschijnen Het schakelen is makkelijk:




door de joy-stick naar voren te houden en de fire-knop in te drukken. Als je de knop indrukt en je doet de joy-stick naar achteren, schakel je terug; dit gaat heel langzaam. Het sturen doe je door links en rechts te bewegen met de joystick. Het is niet mogelijk om op de weg te keren en terug te rijden: hierdoor maak je ook een botsing. Het remmen gaat vrij snel; dit is wel handig als je de andere auto tegenkomt. Je komt maar één auto tegen per race; deze auto doet de vreemdste verdwijnings-trucs. Als die andere auto bij een helling naar beneden rijdt, zie je hem even niet meer. Als je gelijk daarna ook die helling afrijdt, verwacht je die andere auto ook nog daar, maar dit valt zwaar tegen. Een eind verderop is de auto er weer. Misschien heeft die andere auto een raketmotor in plaats van de vier-cilinder 1933 cc motor van de Ford waarin je zelf rijdt.




De bedoeling is ook dat je goed en op tijd schakelt; als je bergen op en af rijdt, moet je in een lage versnelling zitten. Als je over de weg vliegt, krijg je ook schade aan de auto; de schade is te zien op de metertjes boven de versnelling. Het metertje linksboven laat de schade aan de vering zien, het metertje daarnaast duidt de schade van de motor aan, en het metertje linksonder laat zien hoe het met je banden gesteld is. De andere 3 metertjes zijn, aldus de handleiding, alleen interessant voor de kaartlezer. Bij de kaartezer is de kaart te zien: het lichtblauwe gedeelte is wat er gereden is, en het zwarte gedeelte is wat er nog gereden moet worden. Boven de kaart is een klok geïnstalleerd. Dit klokje geeft de tijd aan die er gereden is. De inzittenden van de auto zitten vastgeplakt, zelfs de grootse crash zorgt er niet voor dat ze bewegen, alleen de bestuurder beweegt mooi bij het sturen en schakelen. Dit zorgt er helaas wel voor dat ook het sturen langzaam gaat. Als je tijdens het rijden de poke niet naar voren blijft houden, geeft hij geen gas meer, en sta je stil. Hierdoor moet je de hele weg gas blijven geven, en kan je moeilijk bij een bocht op de weg blijven.

Op de kaart bij de kaartlezer zie je hoe je moet rijden. De bochten zijn er echter eerder dan op de kaart zichtbaar is, en sommige kleine bochten op de kaart zijn er helemaal niet. Deze twee punten zijn heel lastig op donkere- en mistige wegen, omdat je dan een bocht zit te nemen die er niet is, of je neemt geen bocht terwijl die er wel is; dit kan vaak heel pijnlijk zijn voor de bumper.

**Sander Vriesman**

# ST Boeken

*ATARI ST Profibuch*
*door Hans-Dieter Jankowski,*
*Julian F. Reschke en*
*Dietmar Rabich.*
*Düsseldorf, 1988*
*Sybex Verlag*
*ISBN 3-88745-563-0*

*Prijs fl 79,-*
*Beschikbaar gesteld door:*
*Computercollectief*
*Amstel 312*
*1017 AP Amsterdam*

Van de uitgever SYBEX is er onlangs een vernieuwde uitgave verschenen van het ST PROFIBUCH, dat in 1988 in ST nr. 11 al eens de revue is gepasseerd. Omdat dit boek voor veel Atari-hobbisten/programmeurs als een van hun belangrijkste naslagwerken voor de ST-serie wordt beschouwd en omdat de nieuwste uitgave met nogal wat informatie is uitgebreid, leek het ons zinvol dit werk nogmaals onder de aandacht van de lezers te brengen. Het eerste dat opvalt is, dat het een tamelijk dik boekwerk is geworden, dat gebonden is en van een stevige harde kaft is voorzien.
Deze nieuwste uitgave is maar liefst 230 pagina's dikker geworden en telt nu bijna 1000 pagina's. Nu was de vorige uitgave al geen klein boekje en men kon al niet zeggen dat er weinig in stond, maar als daar nog eens 230 pagina's bijkomen, kan men zich voorstellen, dat de schrijvers hebben geprobeerd alle op het moment bekende informatie over de ST-serie in één boek te bundelen, wat ze zeer goed is gelukt. Ik kwam in dit boek informatie en beschrijvingen van routines tegen die ik nog in geen enkel ander boek heb aangetroffen en waarvan ik zelfs het bestaan niet kende.

In tegenstelling tot de vorige uitgave is er duidelijk veel meer aandacht besteed aan een overzichtelijke en uitgebreide index, wat het werken met dit boek erg plezierig maakt.
Alle beschreven systeemroutines zijn voorzien van een declaratie in C, de aanroep in C en assembler, parameterverklaring en eventuele opmerkingen en/of waarschuwingen betreffende het gebruik van deze routines. In het PROFIBUCH zijn alle bekende nieuwe routines van TOS versie 1.4 opgenomen, dat dit jaar nog door Atari op de markt zal worden gebracht. Het grootste deel van de oude informatie is opnieuw bewerkt en uitgebreid, zodat veel nu duidelijker over zal komen. Gedurende de eerdere oplagen van dit boek hebben de schrijvers vlijtig fouten verzameld en het resultaat is nu een (mogelijk) volledig foutloze uitgave, waardoor dit boek tot een standaardwerk voor de ATARI ST-serie mag worden gerekend.
De laatste jaren was er b.v. nagenoeg niets bekend over GDOS met zijn verschillende driver-files, en de opbouw van IMG-files. Ook hierover is nu voldoende informatie aanwezig. Buiten het IMG-formaat worden ook GEM- en het IFF-formaat beschreven. Verder worden de 'negatieve' line-A-variabelen, de DESKTOP.-INF-file, de opbouw van de FAT en de VDI-beeldschermdriver uitvoerig behandeld. Wat de hardware betreft, is het PROFI-BUCH uitgebreid met beschrijvingen over de Blitter, de MEGA-ST-systeembus en de hardware-klok in de MEGA ST's.
Voor iedereen, die dit boek nog niet kent, wil ik hier even vermelden wat het PROFIBUCH in eerdere uitgaven al heeft behandeld, en vanzelfsprekend ook in deze nieuwste oplage. Ten eerste: het bedrijfssysteem en zijn opbouw met zijn specifieke routines (BIOS, XBIOS en GEM-DOS); ten tweede: de VDI-routines met o.a. GDOS met drivers voor beeldscherm, plotter, printer, metafile en kleurpalette; ten derde: de AES-routines die GEM, multitasking, accessories enz. behandelen; ten vierde: de beschrijving van o.a. de gebruikte chips in de ST, het grafieksysteem, soundgenerator en de verschillende uitgangen en ten vijfde: een aanvulling met daarin o.a. informatie over C, Pascal Plus, belangrijke systeem-structuren, karaktersets, XBRA-structuren, het power-up gebeuren, enz..
Voor iedereen die op zijn ATARI-ST iets aan programmeren doet, en dit hoeft niet noodzakelijkerwijs in C of assembler te zijn, maar b.v. ook in GFA-Basic, waarin veel systeemroutines kunnen worden aangeroepen, kan dit boek een prachtig hulpmiddel zijn om meer van uw programma's te maken. Goede dokumentatie is onontbeerlijk en kan veel tijd en moeite besparen, waardoor het plezier in het programmeren alleen maar toeneemt.

**Wil Kerkhof**

---

*Prolog by Example*
*- How to Learn, Teach and Use it*
*Helder Coelho & José C. Cotta*
*Springer-Verlag 1988, 382 pag.*
*Prijs: f 84,85*
*Beschikbaar gesteld door: Verwijs, De Haagse Boekhandel*

Er bestaat duidelijk geen gebrek meer aan inleidingen in Prolog. Het boek dat hier kort besproken wordt, verdient m.i. speciale aandacht. Het is een 'nette' uitgave van het boek How to solve it with Prolog, dat een verkrijgbaar was via het Instituut voor Weg- en Waterbouw van de Universiteit van Lissabon. Toen mij enkele jaren geleden bleek (door navraag bij de K.B. in Den Haag) dat dit boek in geen enkele Nederlandse bibliotheek aanwezig was, heb ik het laten bestellen door een Delftse bibliotheek. U begrijpt het: ik heb er een zekere band mee.
Ik zal enkele kenmerken van het boek geven, vooral door vergelijking van de oude en de nieuwe uitgave. Zoals de nieuwe titel duidelijker aangeeft, wil dit boek Prolog onderwijzen aan de hand van veel voorbeelden.
Het aantal voorbeelden is nu toegenomen van 120 naar 175. Er zitten soms pagina's lange toepassingen bij (vermenigvuldig dit met minstens 5 om het equivalent te krijgen in b.v. Pascal).Het aantal bladzijden is sterk toegenomen: 382 tgv. 212. Toch zit de uitbreiding niet zozeer in het grotere aantal voorbeelden. Een belangrijk verschil is het commentaar dat veel van de programma's nu begeleidt.
Omdat er nu genoeg leerboeken bestaan, zijn de inleidende hoofdstukjes nu weggelaten. Wel wordt nu aandacht aan het leren en ook het onderwijzen van Prolog besteed. Coelho stelt dat de kloof tussen lezen en schrijven van Prolog programma's groter is dan bij de meeste talen. Prolog programma's zijn weliswaar meestal duidelijk (door het declaratieve karakter van Prolog), maar zijn vaak toch zo compact dat hun procedurele betekenis moeilijk te achterhalen kan zijn. Vooral de veelvuldig optredende recursief gedefinieerde predicaten leveren begripsmatige problemen.
Coelho is daarom begonnen om typische probleemoplossingen in Prolog te verzamelen. Het resultaat is een collectie programma's, bijgedragen door een groot aantal onderzoekers op het terrein van het logisch programmeren.
Ieder probleem wordt voorafgegaan door een (vaak zeer) korte omschrijving van het te programmeren probleem. Direct hierna wordt het programma afgedrukt; eventuele commentaren volgen daarna. Ondanks de bijna dubbele omvang van de tekst is er geen sprake van uitvoerig commentaar op de code. Bij een zo wijd scala aan programma's kan dat ook niet.
Bekendheid met de verschillende terreinen wordt verondersteld. Daarom zal waarschijnlijk niemand de neiging hebben om zo'n boek helemaal te doorvorsen, hoewel de toegepaste technieken op het ene gebied wel weer elders van pas kunnen komen.
Merkwaardig genoeg begint deze editie met een nogal felle uithaal tegen Lisp. Iets dgl. ontbrak eerst, en het is niet duidelijk wat hiermee

gewonnen wordt. Waarschijnlijk komt dit door de zeer verschillende programmeerstijlen: functioneel tgv. logisch. Omdat een dergelijke tegenstelling bij b.v. Basic, Pascal en C ontbreekt, laait de discussie op dit punt niet zo hoog op. Aan de andere kant is de lijst met de meest opvallende kenmerken van Prolog is nu weggelaten. Een dergelijke advertentie is nu blijkbaar niet meer nodig.

De bedoeling van het boek is bouwstenen aan te leveren voor eigen toepassingen. Alleen in het eerste hoofdstuk wordt aan algemene tools aandacht besteed (lijstverwerking, bomen, stacks, queues, grafen, sorteren e.d.). Vooraf wordt er nog een lijst gegeven met de 12 'golden rules' voor het schrijven van algemeen begrijpelijke programma's in Prolog.

Om een indruk te geven van de reikwijdte van het boek (en van Prolog) geef ik hier een lijst met de overige onderwerpen: arithmetic, algebra, playing, searching, learning, modeling, designing, drawing, planning, seeing, engineering grammars, constructing personal databases, text formatting, management, building up. Vergeleken met de oude editie zijn enkele hoofdstukken toegevoegd (management, text formatting, drawing), andere zijn uitgesplitst (games, seeing); een enkel hoofdstuk is verdwenen (machine translation). Voor welke grotere toepassingen is Prolog zoal gebruikt in de praktijk? Het antwoord op deze vraag is te vinden in de bijgevoegde thematische lijst, met verwijzing naar de hoofdauteur van het project.

Er is een lijst opgenomen met alle bekende universitaire implementaties (dit zegt veel over de achtergrond van Prolog). Ook is er een lijstje met commerciële producten te vinden.

In de oude editie staat een lijstje met errata: erg nuttig! In deze nette editie past dat natuurlijk niet. Geen fouten? Boeken zonder fouten bestaan helaas niet. Naast de gewone spel-/zetfouten kom je b.v. een vreemde fout tegen als 'Caneghan' i.p.v. 'Caneghem' (o.i.v. Franse uitspraak?), zijn soms woorden weggelaten e.d.

Een merkwaardige weglating in de nieuwe editie is de index. Al hadden ze maar de oude overgenomen! De totaalindruk van deze Springer-uitgave is echter veel beter dan die van de vroegere matrixprinteruitgave.

De literatuurlijst is gelukkig wel bijgewerkt en is aanzienlijk groter geworden zoals verwacht; het officieel ondersteunen van Prolog door het nationale Japanse 'vijfde-generatie project' heeft daar ook aan bijgedragen.

Het gaat hier om een boek vol interessante problemen, oplossingen en de bijbehorende Prolog-code. Een uniek boek, vol met code. Als alles ook nog op schijf te verkrijgen was! Programmeerboeken van een dergelijke spanweidte zijn vrij zeldzaam. Zonder enige reserve aanbevolen als studiewerk naast (of na) een inleidend boek.

**Peter Hendriks**

# OS-9 voor de Atari ST (deel 2)

## 'Het meest flexibele multi-tasking/ multi-user Operating Systeem'

**Twee versies van OS-9/68K**

**Er bestaan twee versies van OS-9 voor de Atari ST, te weten OS-9/68K Personal en OS-9/68K Professional. Hieronder bevindt zich een overzicht van wat beide versies in Duitsland (bij Dr. Keil GmbH) en in Amerika kosten.**

| OS-9/68K | Dr. Keil | In U.S. |
|---|---|---|
| Personal | DM 560,- | $ 150,- |
| Professional | DM 1570,- | $ 600,- |

**Voor het verschil tussen deze twee versies verwijs ik naar de inleiding over OS-9 voor de Atari ST (ST Maart/April nummer 18).**

### TestVersie

Om te beginnen moet ik even wat recht zetten. Ik ben in het vorige verhaal (in ST nr 18) vergeten te vermelden van welke firma de OS-9 versie die ik getest heb, afkomstig was. Bij deze vermeld ik hieronder de naam van het product en de naam van de firma die het levert.

- OS-9/68K Professional voor Atari ST van
  Dr. Rudolf Keil GmbH
  Software-Elektronik-Datentechnik
  Gerhart-Hauptmann-Str. 30
  6915 Dossenheim West-Duitsland

### Oorsprong OS-9

OS-9/68K is een opvolger van OS-9/6809, OS-9 voor machines met een 6809 processor. Veel functies van OS-9 (o.a. floppy format, file managers, device drivers, device descriptors) zijn direct overgenomen uit OS-9/6809, maar voor het grootste deel is het eigenlijk een compleet nieuw besturingssysteem.

### Het besturingssysteem OS-9: de beginselen

Zoals al eerder gememoreerd, is OS-9/68K een multi-tasking/multi- user besturingssysteem. Normaal denkt men dan gelijk aan grote computersystemen, veel en meestal dure hardware (o.a. met harde schijven met grote opslag capaciteiten). Maar OS-9 is (ook) geschikt voor kleinere systemen, en ook voor micro's als een Atari ST.

### Multi-Tasking/Multi-Processing

Multi-Tasking (besturings)systemen zijn in staat om meerdere processen op een enkele CPU (Central Processing Unit = Centrale Verwerkings Eenheid) quasi parallel uit te voeren. Een computersysteem met één processor kan maar één enkel proces uitvoeren. Andere processen die 'gelijktijdig' op dit systeem moeten lopen, kunnen op dat moment niet werken. De rekentijd van de CPU moet derhalve onder alle lopende processen verdeeld worden. Dat betekent dat delen van de afzonderlijke processen snel wisselend na elkaar uitgevoerd worden en na korte tijd weer afgebroken worden om door andere processen vervangen te worden. Wordt een proces opnieuw opgeroepen, dan gaat zijn uitvoering verder op de plaats waar het programma eerder onderbroken werd.
Meestal wordt aan alle processen een vaste tijd toegekend waarin zij aktief kunnen zijn. Deze hoeveelheid tijd wordt ook wel 'time slice' genoemd. Aan het begin van zo'n 'time slice' krijgt een proces de controle over het systeem toebedeeld. Aan het eind van deze tijd wordt de controle over het systeem aan het proces onttrokken en toebedeeld aan een ander proces.

### Wachtrijen

De volgorde waarin de 'time slices' aan de verschillende processen toebedeeld worden moet natuurlijk op de een of andere manier geregeld worden.
Hiervoor worden de processen in wachtrijen (queues) gezet. Het proces dat zich aan het hoofd van zo'n rij bevindt, krijgt bij het begin van een nieuwe 'time slice' de besturing en dus de controle over het systeem. Aan het eind van een 'time slice' wordt het proces weer onderbroken (suspended) en aan het eind van de wachtrij geplaatst. Dit geldt ook voor nieuw gestarte processen. Dit manipuleren van de processen in het systeem wordt gedaan door een zogenaamde 'scheduler', die zich in het binnenste van het besturingssysteem bevindt.
Deze 'scheduler' wordt pas aktief, als er een interrupt binnenkomt van externe hardware, of als er een 'time slice' verstreken is. Zolang er niet zo'n interrupt binnenkomt, houdt het huidige proces de besturing over het systeem totdat z'n 'time slice' voorbij is.

### Voorzorgsmaatregelen

Diegenen die wel eens in assembler geprogrammeerd hebben, weten dat de kleinste fouten ertoe kunnen leiden dat het systeem gaat 'hangen', niets meer doet!
Wat in single-tasking systemen alleen maar ergerlijk is, kan in multi-tasking en evt. multi-user systemen fatale gevolgen hebben: andere gebruikers zullen het niet echt leuk vinden als een of andere zot ervoor verantwoordelijk is dat het hele systeem gaat 'hangen'. Ook als er alleen maar meerdere taken tegelijk draaien is het handig als deze taken niet in elkaars vaarwater gaan zitten, d.w.z. elkaar data/code gebieden niet beïnvloeden.
Als het systeem 'down' gaat, wordt dit meestal veroorzaakt door veranderingen in de systeem-stack. Daarom maakt men in besturingssystemen als OS-9 onderscheid tussen een gebruikers-stack en een systeem-stack. De gebruiker mag niet in staat zijn de systeem-stack te veranderen. Daarom kunnen sommige instructies alleen in 'system mode' - de 68000-processor kent een 'user mode' en een 'system mode' -uitgevoerd worden. Bij iedere 'system call' wisselt de processor van 'user mode' naar 'system mode' om deze uit te voeren en weer terug. Het besturingssysteem werkt zo altijd met de systeem-stack. Wanneer de processor weer in de 'user-mode' is teruggekeerd kan alleen nog van de gebruikers-stack gebruik gemaakt worden.

### Prioriteiten

Alle processen in het systeem hebben een bepaalde prioriteit. Deze prioriteit wordt toegekend bij het starten van een proces en deze kan veranderen tijdens de executie van een proces. Om er voor te zorgen, dat sommige processen vaker aan de beurt komen, dan andere proces-

sen, wordt aan deze processen een hogere prioriteit toegekend dan aan anderen.
In de boven geschetste wachtrij worden de processen in volgorde van prioriteit achter elkaar opgeslagen. Het proces met de hoogste prioriteit bevindt zich aan het hoofd van de wachtrij. Wanneer er een 'time slice' afloopt, worden alle prioriteiten van alle in de wachtrij aanwezige processen met één verhoogd. Een proces dat aan het van een 'time slice' wordt onderbroken, wordt opnieuw op de juiste plaats (gezien zijn prioriteit) in de wachtrij opgenomen.

## Synchronisatie

Vaak zijn er toepassingen die een of andere synchronisatie tussen verschillende processen vereisen. Hierbij wordt de executie van een proces pas voortgezet, indien een ander proces in een bepaalde toestand gekomen is. De synchronisatie tussen processen wordt mogelijk gemaakt d.m.v. 'signals', die een proces naar een ander proces kan zenden. Hierdoor kan men de processen in een drietal categorieën onder verdelen:

1) Het proces is 'active', d.w.z. het proces is klaar om uitgevoerd te worden. Het bevindt zich in de wachtrij van actieve processen en wordt zodra het aan het hoofd van deze wachtrij is aangekomen geactiveerd voor een periode van een 'time slice'.

2) Het proces is 'waiting', d.w.z. het proces wacht op een door hem gecreëerd proces. Dit proces zal op een zeker moment het wachtende proces een signaal geven, dat hij klaar is met zijn executie. Hierdoor wordt het wachtende proces weer 'active'. Het proces bevindt zich in de wachtrij van wachtende processen en zodra dit een seintje krijgt dat hij niet meer hoeft te wachten, aan de voorwaarde(n) waarop dit proces wachtte is voldaan. Het proces wordt dan in de wachtrij van actieve processen gezet.

3) Het proces is 'sleeping', d.w.z. het proces wacht bijvoorbeeld totdat een door opgedragen input/output (I/O) operatie is beëindigd.

Uit de 'waiting' en 'sleeping' toestand kan een proces in de 'active' toestand geraken, doordat een ander proces hem een bepaald signaal geeft. Stel dat bijvoorbeeld een proces A aan het wachten is totdat een parallel lopend proces B een bepaalde functie heeft uitgevoerd. Het proces A bevindt zich dus in de 'sleeping' toestand. Als proces B deze functie heeft uitgevoerd, dan zend dit proces een 'Wake-Up' signaal naar proces A, waardoor proces weer 'active' wordt. Als door proces A een proces B gestart wordt, dan wordt proces A automatisch in de 'waiting' toestand gezet en is proces B 'active'. Proces A wordt pas weer in de 'active' toestand gezet, indien proces B klaar is met zijn executie.

## Geheugen Organisatie

De verschillende quasi-parallel-lopende programma's of processen moeten naast de CPU(-tijd) ook het geheugen van het system onderling verdelen. Daarom moet de verdeling van dit geheugen over de verschillende processen door het besturingssysteem geregeld worden.
Van belang hierbij is zoveel mogelijk het gebruik van absolute adressen te vermijden, omdat programma's die hier gebruik van maken alleen onder bepaalde omstandigheden gedraaid kunnen worden. Om zoveel mogelijk problemen te voorkomen, probeert OS-9 de programmeur zoveel mogelijk 'relocatable' code te laten schrijven. Dit is code, die op iedere plaats in het geheugen geladen kan worden. OS-9 zoekt bij het laden van zulke programma's wel uit waar zij geladen gaan worden. Het is daarom ook niet toegestaan absolute sprongen of absolute geheugen operaties uit te voeren in een programma. Bij het programmeren moet men zich beperken tot de relatieve adresseringsmethoden van de 68000. Hierdoor ontstaat tegelijkertijd een beperking op de grootte van programma's: bij de 68000 is het argument bij relatieve adressering een woord groot (16 bits). Hieruit kan men een adresseringsbereik van 32Kbyte in positieve en in negatieve richting afleiden. Deze beperking kan men echter d.m.v. 'jump tables' (of vector tabellen) omzeilen.

## Als twee gebruikers hetzelfde doen ...

In een muti-user systeem komt het vaak voor, dat verschillende gebruikers met een en hetzelfde programma willen werken. In dit geval zou het verspilling van geheugenruimte zijn, indien het het programma voor iedere gebruiker apart in het geheugen geladen zou worden.
OS-9 doet dit dan ook niet en laadt dit programma maar één maal. Het kan dan door alle gebruikers aangeroepen en gebruikt worden. Om te verhinderen, dat de gebruikers elkaar in de weg zitten, moeten een aantal zaken worden opgelost.
Gebruiker A werkt bijvoorbeeld met een tekstverwerker. Hierbij wordt de ingevoerde tekst, de cursor-positie etc. in het geheugen onthouden. Als de 'time slice' van dit proces voorbij is en gebruiker A nog niet klaar is met het invoeren van tekst, dan kan bijvoorbeeld gebruiker B voor een bepaalde tijd de tekstverwerker aan de slag. Wederom worden bovengenoemde data veranderd. Als de gebruikte tekstverwerker maar een data gebied zou bezitten, dan zou gebruiker B de data van gebruiker B aantreffen en deze veranderen. Om het mogelijk te maken dat meerdere gebruikers met een en hetzelfde programma kunnen werken, moet iedere gebruiker dus zijn eigen datagebied hebben.
Verder moet het programma (b.v. de tekstverwerker) 'reentrant' opgebouwd zijn, d.w.z. het programma moet oneindig vaak opgeroepen kunnen worden, ook na een onderbreking van zichzelf. De code waaruit dit programma bestaat mag niet veranderen, als dit programma uitgevoerd wordt. Het tegenovergestelde van 'reentrant' is 'self-modifying' code, d.w.z. code die zichzelf verandert indien deze uitgevoerd wordt.
Als men de data van een programma, nodig voor iedere gebruiker, binnen een programma opslaat, dan zal de resulterende code ook niet 'reentrant' maar 'self-modifying' zijn, omdat een gebruiker bij het opnieuw betreden van dit programma andere data aan kan treffen, dan de vorige keren. Het is dus niet mogelijk 'self-modifying' programma's quasi-parallel door meerdere gebruikers uit te laten voeren.

## OS-9 in detail

Het besturingssysteem OS-9 is opgebouwd uit verschillende delen, modules geheten. Deze modules zijn eigenlijk niets anders, dan een manier voor het samennemen van data. De modules moeten er op een bepaalde manier uitzien. Hoe deze er dan uitzien, zal later in dit verhaal duidelijk worden. Het wordt gevormd door de kern van het systeem (Kernel), de file managers voor sequentiële I/O (SCF = Sequential Character File Manager), voor random-block-I/O (RBF = Random Block File Manager), voor sequential block I/O (SBF = Sequential Block File Manager), communicatie tussen processen (PipeMan = Pipe File Manager), en door meerdere device-driver-modules en device-descriptor-modules. Device-driver modules en device-descriptor-modules horen bij een bepaalde file-manager. Bij de kern van het systeem horen bovendien nog de modules StClk (de klok van het systeem) en Init.

Van boven naar beneden kunnen de volgende 4 lagen onderscheiden worden:

```
laag 0: system modules          |
laag 1: file managers      ---  |
laag 2: device drivers     ---  | --> I/O
                                |     systeem
laag 3: device descriptors ---  |
```

**FIGUUR 1**

Naast de modules uit bovenstaande figuur bevat OS-9 nog een ander soort module, Trap-Handlers geheten. Voorbeelden van deze Trap-Handler-Modules zijn bijvoorbeeld CIO en Math. Alle I/O operaties uit C-programma's (alle commando's) lopen via deze Trap-Handler. Deze geeft ze weer verder door aan het systeem en in het byzonder aan de Kernel. Alle mathematische operaties lopen via de Trap-Handler Math. Deze berekent de opgegeven operatie zelf.

De module Kernel of kortweg de kernel van OS-9 is verantwoordelijk voor alle I/O-operaties, het beheer van het geheugen, het sturen van de multi-tasking en het aanroepen van andere besturingssysteem modules als StClk en Init. Alle I/O operaties worden door de kernel doorgesluisd naar de file-manager, die verantwoordelijk is voor deze I/O operaties. Zo is de file-manager PipeMan er voor de communicatie tussen de verschillende processen. Onder deze file manager bevinden zich de device-driver Null en de device-descriptor Pipe. Als er communicatie is tussen twee processen zorgt de kernel ervoor dat de juiste delen van PipeMan aangeroepen worden, zodat deze de device-driver Null opdracht kan geven de gevraagde I/O-operatie uit te voeren.

De file-managers SCF, SBF en RBF verbinden de kernel met het eigenlijke I/O-systeem. De eigenlijke I/O-operaties worden uitgevoerd door de respectievelijke device-drivers. Zo bevindt zich onder SBF het device mt0 (device-descriptor mt0) voor het werken met bijvoorbeeld een tape streamer, onder SCF het device p (device-descriptor p) als naam voor een printer en de devices term en t1 als de namen voor twee terminals. Het device p wordt bestuurd door de device-driver stptr, de devices term en t1 door stcio en stsio. Onder RBF bevinden zich het device d0 als naam voor de 'eerste' floppy-drive en h0 als naam voor de 'eerste' harde schijf, en het device r0 als naam voor een zogenaamde ram-disk. De floppy-drive en de harde schijf worden beide door dezelfde device-driver, stfd geheten aangestuurd. De ram-disk wordt 'bestuurd' door de device-driver RAM. De namen van de verschillende devices zijn in de verschillende device-descriptor-modules gedefinieerd.

In multi-tasking besturingssystemen als OS-9 moet het mogelijk zijn om programma's na een bepaalde hoeveelheid tijd te kunnen onderbreken om meerdere processen quasi-parallel te kunnen uitvoeren. Dit 'onderbreken' wordt verzorgd door de module StClk, de klok van het besturingssysteem. Deze klok geeft bijvoorbeeld iedere 10ms een interrupt, zodat de scheduler van het besturingssysteem weet wanneer hij een nieuw proces (misschien ook een ander proces) moet starten.

## Opbouw van de Modules

Zoals nu duidelijk moge zijn, bestaat het besturingssysteem OS-9 uit een behoorlijk aantal modules, die voor verschillende dingen verantwoordelijk zijn. Ook gebruikers-programma's, utilities etc. zijn natuurlijk in wezen modules. Om modules van elkaar te onderscheiden en om modules te kunnen herkennen is een module op de volgende manier opgebouwd:

Module Header

Hier bevindt zich de informatie over de naam van de module, zijn toepassing en de soort data dat de module bevat. Op het moment zijn de volgende soorten modules gedefinieerd:

| | |
|---|---|
| * program | : gebruikersprogramma's en commando's |
| * subroutine | : sub-programma's |
| * data | : data modules |
| * traplibrary | : modules voor het gebruik van traps |
| * system | : OS-9 system module, als de Kernel, StClk en Init |
| * filemanager | : voor I/O, als SCF, SBF, RBF en PipeMan |
| * devicedriver | : device drivers voor floppy disks, printers en terminals (stfd, stptr, stsio, stcio) |
| * devicedesc | : definitie van een device (d0, p, term ...) |

Verder bevindt zich in de header van een module o.a. informatie over in wat voor taal (assembler, Pascal etc.) de module geschreven is.

Module Body

Hier bevindt zich de eigenlijke inhoud van de module. Het kan hierbij om code of data gaan.

Cyclic Redundancy Check (CRC)

M.b.v. deze waarde, die zich aan het eind van iedere module bevindt, kan iedere keer als de module gebruikt gaat worden onderzocht worden of de module nog wel in orde is.
Als de CRC niet klopt met de module, dan geeft het besturingssysteem een melding dat de betreffende module corrupt is en op dit moment niet (meer) gebruikt kan worden, zodat geen verdere schade kan optreden. Middels een aantal utilities kan men de CRC van een module repareren (opnieuw laten berekenen en invullen).

## Initialisatie van OS-9

Om iets men een module te kunnen doen moet deze module eerst door het systeem in het hoofdgeheugen geladen worden. Na een system-reset moeten dus de modules van het besturingssysteem beschikbaar zijn. Hiervoor bestaan twee mogelijkheden: het systeem wordt van floppy-disk of harde schijf geladen of de modules van het besturingssysteem bevinden zich in ROM.

In het eerste geval bevindt zich op de floppy (of harde schijf) een speciaal bootfile, dat alle (systeem) modules bevat. Alle soorten modules, dus ook programma's kunnen zich in ROM of in het bootfile bevinden. Een bootroutine initialiseert het hoofdgeheugen, alloceert de juiste hoeveelheid geheugen voor het bootfile en laadt het systeem in RAM. Aansluitend wordt de Kernel van OS-9 gestart. In het tweede geval bevinden zich alle modules al in het geheugen (in ROM). Het systeem wordt geinitialiseerd en de Kernel wordt gestart.

De Kernel begint dan met zoeken van modules. Deze kunnen zich in RAM en ROM bevinden. Herkend worden de modules, waarbij de module header correct is en de CRC ook. De gevonden modules worden aan de zogenaamde module-directory toegevoegd (zie fig. 1, ST 18). Deze module-directory bevat informatie over deze modules, nl. de naam van de module, het type van de module, de grootte van de module, de 'link-count' etc.

## Het laden van modules

Met het OS-9 commando LOAD kan men modules van bijvoorbeeld floppy laden in het hoofdgeheugen van het systeem. De geladen module wordt eveneens aan de module directory toegevoegd.
Als een programma's aangeroepen wordt, dan wordt allereerst gekeken of dit programma zich al in de module-directory bevindt. Wanneer dit niet het geval is, wordt een poging gedaan de gevraagde module van schijf te laden. Bevindt zich de gevraagde module op schijf, dan wordt deze geladen.
Als men nu als maar modules aan het laden is, dan vult zich het hoofdgeheugen langzamerhand met modules, totdat er geen geheugen meer vrij is (dit is haast niet mogelijk, ofschoon dit afhankelijk is van de grootte van het beschikbare geheugen). Ook modules, die men bijvoorbeeld niet meer wenst te gebruiken bevinden zich dan nog in het geheugen. Om te verkomen dat dit alles gebeurt, kan men de eens geladen modules ook weer uit het geheugen weghalen, zodat er weer (meer) geheugenruimte vrij komt.
Bij iedere module in de module-directory hoort een zogenaamde 'link count'. Bij iedere aanroep van een module wordt de link count van deze module met een verhoogd. Is de module niet meer nodig, dan wordt de link count weer met een verlaagd. Wordt de link count toevallig 0, dan wordt de module uit het geheugen verwijderd en het geheugen dat de module in beslag nam, wordt aan het systeem teruggegeven. Verder wordt de link-count van een pas geladen module op 1 gezet.
Deze link-count maakt tegelijk een eenvoudig beheer mogelijk van modules, waarvan meerdere gebruikers tegelijk gebruik van kunnen maken. Stel dat meerdere gebruikers met dezelfde editor willen werken. Gebruiker A roept de editor aan. Stel, deze bevindt zich nog niet in het geheugen. De editor wordt geladen van schijf, de module wordt toegevoegd aan de module-directory en de link-count van de module wordt op een gezet. Bovendien wordt voor gebruiker A een stuk locaal geheugen gealloceerd voor de opslag van zijn eigen data. Gebruiker B wenst ook met de editor te werken en roept deze aan. Deze bevindt zich al in de module-directory en kan daarom direct aangeroepen worden. De link count van de module wordt nu met een verhoogd tot de waarde 2. Ook voor gebruiker B wordt een stuk locaal geheugen gealloceerd voor zijn eigen data. Gebruiker A stopt na enige tijd met werken en verlaat de editor. De link-count wordt verlaagd en het locale geheugen van A wordt weer vrijgegeven. De editor blijft nog in het geheugen. Pas als gebruiker B zijn werk met de editor beëindigd wordt de link-count op 0 gezet en de editor uit het geheugen gehaald.
De link-count van een module in de module-directory kan met een verhoogd worden middels het commando LINK en met een verlaagd worden middels het commando UNLINK. Zo kan een module toch in het geheugen blijven, ondanks dat deze door geen enkele gebruiker gebruikt wordt.

## Het zoeken van modules

Om modules in het geheugen te kunnen vinden, moet OS-9 een module kunnen herkennen en kunnen onderscheiden van andere data, die zich in het geheugen bevindt. OS-9 moet zo snel mogelijk en met de grootst mogelijke zekerheid een module in het geheugen kunnen herkennen. Omdat onder OS-9 bijna alle even adressen in aanmerking komen als startadres van een module zou het steeds opnieuw controleren van de CRC eenvoudig weg te veel tijd in beslag nemen. OS-9 herkend een module aan de eerste twee bytes van een module, de zgn. sync-bytes. Deze hebben een vaste waarde, nl. $4AFC, een waarde die in het geheugen - in code of data modules - relatief weinig voorkomt. Bovendien ziet de 68000 deze twee bytes nooit als een instructie. Dus als men bij het zoeken naar modules deze twee bytes aantreft, dan kan men

er behoorlijk zeker van zijn, dat men het begin van een module gevonden heeft.

Toch kunnen deze twee bytes nog iets anders betekenen, bijvoorbeeld een datum. Dus er moet nog meer gedaan worden om met zekerheid te kunnen zeggen, dat een module gevonden is. Hiervoor wordt over de gehele module-header een pariteitswaarde berekend en aan het einde van de header bewaard. Nu kan het systeem snel nagaan de header correct is door de pariteit over de header te berekenen en deze te vergelijken met de in het geheugen aanwezige pariteitswaarde van de (evt.) module.

Deze methode levert een sn elle manier van testen of een module aanwezig is of niet, omdat het behoorlijk onwaarschijnlijk is, dat in het geheugen toevallig een stuk data aanwezig is dat aan beide bovengenoemde eisen voldoet. Maar als twee bits op dezelfde positie toevallig van toestand veranderd zijn (0 in 1 of v.v.), dan geldt de pariteits-waarde nog steeds. Om helemaal zeker te zijn, wordt over de hele module een 24-bits CRC-check berekend. Als deze ook klopt met de in het geheugen aanwezige CRC-waarde, dan is het practisch uitgesloten, dat een toevallige stuk data herkend wordt als zijnde een module.

## Informatie in de Module-Header

In de header van een module zijn verschillende parameters aanwezig, die of door OS-9 of door de module zelf gebruikt worden. In onderstaande fig. 2 is de betekenis van de verschillende parameters nader uitgelegd.

Aan de linker kant staat een offset, nl. het relatieve adres van een veld of parameter rekenend vanaf het begin van de module. In de volgende kolom staat de naam van de betreffende parameter, zoals deze door de assembler onder OS-9 gebruikt wordt. De laatste kolom geeft een korte beschrijving van de functie van de verschillende parameters. Sommige van deze parameters zullen we wat nader beschrijven:

MSSize : bevat de totale grootte van de module, inclusief header, body en CRC waarde.

M$Owner : bevat een code voor de eigenaar van de module. In multi-user systemen is dit handig om te voorkomen, dat bepaalde gebruikers een bepaalde module onrechtmatig zouden gebruiken.

M$Name : bevat een offset naar de naam van de module, zoals deze zich in de module-directory bevindt, en waarmee de module ook aangeroepen en gebruikt kan worden. Dit is een offset naar de plaats waar de naam van de module staat, omdat de naam van een module niet vaststaat en een steeds varierende lengte kan hebben.

M$Accs : bevat een code voor de manier, waarop de eigenaar en de rest van het systeem (public) gebruik kunnen maken van de module. Hiermee kan men de module het bijvoorbeeld mogelijk maken, dat de module alleen door de eigenaar uitgevoerd mag worden.

M$Type : bevat het type van de module. Zoals al eerder beschreven is kan dit veld de volgende type aanduidingen bevatten:
Program, SubRoutine, Data, TrapLibrary, System, FileManager, DeviceDriver en DeviceDescriptor.

M$Lang : bevat een aanduiding van de taal waarin de module geschreven is.

M$Attr : bevat een code die ondermee aangeeft of de module reentrant is, zodat al of niet meerdere gebruikers deze module tegelijk kunnen gebruiken.

M$Revs : bevat een revisienummer. Als men een module probeert te laden, die zich al in de module-directory bevindt, dan wordt deze module alleen geladen, indien het revisienummer van de nieuwe module hoger is dan dat van de oude (die zich al in het geheugen bevindt). De oude module wordt als het ware vervangen door een nieuwe module met dezelfde naam. Hierdoor is het ten allen tijde mogelijk modules in ROM te 'vervangen' door modules in RAM met een hoger revisienummer.

M$Edit : bevat het editie nummer, een code voor het versienummer van de software. Dit is handig voor de programmeur, omdat hij/zij dan kan weten met welke versie hij/zij van de module aan het werken is. Dit veld wordt door het systeem niet benut.

M$Parity : bevat de pariteitswaarde berekend over de module- header.

Afhankelijk van het type van de module kunnen nog additionele parameters in de header van een module opgenomen worden, te weten

**Additionele Header Velden**

| Offset | Naam | Betekenis |
|---|---|---|
| -$30 | M$Exec | offset naar het start-adres van het programma |
| -$34 | M$Excpt | offset naar een trap-routine |
| -$38 | M$Mem | benodigde geheugenruimte |
| -$3C | M$Stack | vereiste stackgrootte |
| -$40 | M$IData | offset naar de geïnitialiseerde data |
| -$44 | M$IRefs | offset naar referentietabellen |
| -$48 | M$Init | offset naar initialisatieroutine |
| -$4C | M$Term | offset naar terminatieroutine |

M$Exec : bevat een offset naar het start-adres van de module. Samen met het eigenlijke start-adres van de module (bepaald tijdens het laden van de module) bepaalt deze offset waar de eigenlijke executie van een module begint (welke instructie het eerst uitgevoerd moet worden).

M$Excpt : bevat een offset naar een trap-routine, die niet geïnitialiseerde user-traps afhandelt. Voor een beschrijving van traps bij een 68000 kan men het beste een boek over 68000-assembler erbij pakken.

M$Exec en M$Excpt worden gebruikt modules van type Program, TrapHandler, DeviceDriver, FileManager en System.

M$Mem : deelt het systeem mede, hoe groot het datagebied is van de module.

M$Mem wordt gebruikt in modules, die hun eigen datagebied moeten alloceren, te weten Program, TrapHandler en DeviceDriver modules.

M$Stack : bevat de hoeveelheid stackruimte die de module nodig heeft. Dit moet er vooral voor zorgen, dat de verschillende modules elkaar niet in de weg gaan zitten. De grootte varieert van programma tot programma en mag in ieder geval niet gelijk zijn aan 0.

M$IData : bevat een wijzer naar het gebied waar zich de door het systeem toegewezen data bevindt. Dit gebied bevat in het begin geen zinnige waarden, maar uit datagebieden uit de module- body kan geïnitialiseerde data gekopieerd worden naar dit gebied. Dit wordt automatisch gedaan door de linker.

**FIGUUR 2**

| Standard | Header | Velden |
|---|---|---|
| Offset | Naam | Betekenis |
| $00 | M£ID | sync-bytes ($4AFC) |
| $02 | M$SysRev | System Revision Level ($0001) |
| $04 | M$Size | grootte van de module |
| $08 | M$Owner | eigenaar van de module |
| $0c | M$Name | offset naar de naam van de module |
| $10 | M$Accs | access permissions |
| $12 | M$Type | type van de module |
| $13 | M$Lang | taal waarin module geschreven is |
| $14 | M$Attr | attributen |
| $15 | M$Rev | revisienummer |
| $16 | MSEdit | editienummer |
| $18 | M$Usage | offset naar commentaar over de module |
| $1C | M$Symbol | offset naar de symbol-table |
| $20 | Reserved | |
| $2E | M$Parity | pariteitswaarde over de module-header |
| $30-up | | module afhankelijke parameters |
| | Module Body | |
| | CRC-waarde | |

M$IRefs : bevat een offset naar een tabel, die het mogelijk maakt absolute wijzers in code en data gebieden te kunnen gebruiken. De wijzers in deze tabel moeten namelijk afhankelijk van het startadres van de module en het startadres van het data gebied gecorrigeerd worden. Bij de aanroep van een module wordt deze correctie automatisch uitgevoerd.

M$Stack, M$IData en M$IRefs worden alleen in Program- en TrapHandler-modules gebruikt.

M$Init en M$Term komen alleen voor in TrapHandler-modules en dienen er resp. voor om de trap-routines uit deze TrapHandlers aan het systeem toe te voegen en uit het systeem te verwijderen.

Tot zover iets over de header van een module en zijn functie. Het zou te ver voeren om precies uit te leggen hoe de verschillende modules eruit zien en hoe men deze zou moeten schrijven. Mensen die hierin geïnteresseerd zijn en dit zelf wel eens zouden willen doen kunnen of zelf OS-9 aanschaffen of het bij bijvoorbeeld een kennis proberen (liever niet bij mij). Eigenlijk zou het wel aardig zijn om contact te krijgen met andere OS-9 gebruikers om ervaringen en eventueel software te kunnen uitwisselen.

## 'OS-9 System Calls'

Wat zijn dit eigenlijk? Als men bijvoorbeeld een schrijfoperatie wil doen en men wil dat programmeren in 68000-assembler, dan dient men de OS-9-system call I$Write uit te voeren. Onderstaand voorbeeld geeft aan hoe:

```
...
lea       String(pc).a0    ; laat a0 wijzen naar de string String
move.w    # 1.d0           ; pad nummer 1 == standard output
move.7    # StrLen,d1      ; zet de lengte van de string in d1
OS9       I$Write          ; OS-9 system call I$Write
...
```

wat eigenlijk overeenkomt met

```
...
lea       String(pc),a0    ; laat a0 wijzen naar de string String
move.w    # 1,d0           ; pad nummer 1 == standard output
move.7    # StrLen,d1      ; zet de lengte van de string in d1
trap      # 0              ; OS-9 system-call loopt via trap #0 van de 68000
dc.w      I$Write
...
```

OS-9 bevat heel wat van deze systeem aanroepen, die allen op bovenstaande manier gebruikt kunnen worden. Deze systeem aanroepen worden uitgevoerd in 'system state', d.w.z. de 68000-processor bevindt zich in 'system state'. Normaal bevindt zich de processor tijdens de uitvoering van een programma in de 'user state'. Zodra in een programma een systeemaanroep wordt gedaan, wisselt de processor van 'user state' naar 'system state'. Als de systeem aanroep beëindigd is, wisselt deze weer terug naar 'user state'.

Het belangrijkste verschil tussen 'user state' en 'system state' is het volgende: in 'user state' kan een gebruikersprogramma alleen bij de data van het systeem d.m.v. een systeemaanroep. Als de processor zich in 'system state' bevindt doordat een gebruikersprogramma een systeem aanroep heeft gedaan, dan kan dit programma niets vernielen aan de data van het systeem, omdat dit programma daar geen toestemming voor heeft, omdat dit programma oorspronkelijk in 'user state' gestart was. Voor verdere verschillen tusen 'user state' en 'system state' verwijs ik naar een boek over 68000-assembler.

Een systeem aanroep binnen OS-9 komt overeen met het uitvoeren van een trap-instructie (trap 0, zie voorbeeld hierboven). Welke aanroep er precies volgt na de trap 0-instructie wordt bepaald door een woord (16-bits) waarde die direct volgt op de deze instructie (dc.w I$Write, met I$Write een waarde die aangeeft dat een output-operatie gedaan dient te worden). Een systeem aanroep wordt afgesloten met een RTE-instructie (Return From Exception). Het uitvoeren van een OS-9 call zorgt ervoor dat een bepaald gedeelte van de kernel van OS-9 wordt uitgevoerd. Welk gedeelte wordt bepaald door de parameter van de macro OS9.

Wenst men meer te weten over deze systeem aanroepen, dan dient men in een boek over 68000-assembler (traps etc.) te kijken en evt. in een OS-9 Technical Manual. Alleen het gebruik van traps op een 68000 is al de moeite van het bekijken waard.

## De module INIT

OS-9 bezit dankzij zijn modulaire structuur een zeer grote flexibiliteit. Dit zou niet van toepassing kunnen zijn op de kernel van het systeem, indien deze niet makkelijk te configureren zou zijn. Alle systeem-afhankelijke parameters die de kernel nodig heeft bij het starten van het systeem bevinden zich in één module, namelijk de module INIT. Deze module bevat geen uitvoerbare code, maar alleen data, nl. een tabel met (systeem-afhankelijke) parameters.

De meeste parameters hebben betrekking op de grootte van systeemtabellen. Verder is hier het aantal randapparatuur (terminals, printers etc.) vastgelegd. Ook bevinden zich hier de naam van het console van het systeem, de naam van de root-directory van het device waarvan evt. geboot kan worden, de naam van de klokmodule en de naam van het proces dat als eerste door de kernel gestart dient te worden (meestal sysgo geheten). Verder vindt men hier ook enkele parameters voor het gebruik van het verdelen van de beschikbare systeemtijd 'time sharing' en van het beheren van de processen 'proces scheduling'.

## De module SYSGO

Sysgo is het eerste programma dat na het starten van het systeem door de kernel uitgevoerd wordt. Deze probeert bijvoorbeeld de directory CMDS op de het 'boot device' (zie INIT) te openen. In deze directory bevinden zich normaal alle commando's en utilities van OS-9. OS-9 zoekt ook bij het zoeken naar zo'n commando op schijf het eerst in deze directory. Vervolgens probeert sysgo de shell (SHELL geheten) te starten, die dan verder de startup procedure afmaakt.

De shell SHELL is de 'command line interpreter' (commando regel interpretator), verantwoordelijk voor het analyseren en uitvoeren van commando's gegeven door gebruikers.

In de rest van de startup-procedure probeert de SHELL een file met de naam startup uit te voeren. Hierin staan commando's als het zetten van de systeem tijd (middels setime -s = haal tijd van de real-time klok) en bijvoorbeeld het starten van wat terminals (middels b.v. tsmon /term = start 'time sharing monitor' op /term). Ook de initialisatie van randapparatuur kan hierin voorkomen, als mede het creëren van een ram-disk.

Met de modules INIT en SYSGO kan men OS-9 dus eenvoudig aanpassen aan verschillende hardware-configuraties.

## Systeem Geheugen

OS-9 heeft de beschikking over een globaal stuk systeemgeheugen van 8 Kbyte. Hier zijn verschillende variabelen voor het beheer van het systeem ondergebracht, bijvoorbeeld wijzers naar blokken geheugen, wijzers naar procesdescriptors en systeem-parameters als het type van de processor dat gebruikt wordt (68000, 68010 etc.). Men kan dit gebied direct adresseren, lezen en schrijven, maar om de voorkomen dat dit gebied vernield wordt kan men beter gebruik maken van een OS-9 systeemaanroep, nl. F$SetSys. Deze mag alleen door de superuser van het systeem gebruikt worden (superuser = user met user id (0,0) als id voor (groupid,userid)).
Met deze systeemaanroep kunnen alle globale variabelen gelezen worden, maar maar enkele kunnen er ook daadwerkelijk veranderd worden. Het systeem is dus in staat om verschillende programma's (modules) te laden en uit te voeren.

## Starten van een Proces

Als een proces aangeroepen wordt en evt. gestart, moet dit proces gecontroleerd kunnen worden door het systeem. Hiervoor moet het systeem over informatie van alle afzonderlijke processen beschikken. Voor ieder proces wordt daarom een blok met informatie gereserveerd (proces descriptor), dat alle belangrijke informatie over het proces bevat: de naam van het proces, de prioriteit van het proces, de owner, de I/O paden etc. Deze proces descriptors worden in de al eerder genoemde wachtrijen geplaatst afhankelijk van de toestand van een proces.
Het eigenlijke aanroepen of starten van een proces gebeurt d.m.v. de systeemaanroep F$Fork. Hiermee kan een proces een sub-proces (ook wel kind genoemd) starten, dat dan onafhankelijk van dit proces (ook wel ouder genoemd) uitgevoerd gaat worden.
Verder moet voor het nieuwe proces geheugen gealloceerd worden en moeten de diverse I/O-paden doorgegeven worden. Dit gebeurt door aan F$Fork de juiste parameters mee te geven. In onderstaand voorbeeld wordt bijvoorbeeld een sub-proces gestart met de naam mdir:

```
.....       Param(pc),a1      ; a1 wijst naar de parameters van het sub-proces
lea           ParamLen,d2     ; lengte van de parameter string ==> d2
move.7        50,d4           ; prioriteit van sub-proces = 50 ==> d4
move.w      ork               ; subroutine Fork
bsr.s
.....       F$Wait            ; wacht totdat sub-proces klaar is met z'n executie
OS9
ForkErr     F$Exit            ; einde programma, exit
OS9

* Subroutine Fork

Fork
lea         Name (pc),a0      a0 wijst naar de naam van het sub-proces
clr.w       d0                ; type/revision veld = 0
clr.7       d1                ; geen extra geheugen ruimte nodig
move.w        3,d3            3 I/0 paden
OS9         F$Fork            ; start process
Bcs         ForkErr           ; fout opgetreden, spring naar ForkErr
rts

Name        dc.b              'mdir',0 ; naam van sub-proces
Param       dc.b              '-e',C$CR ; parameter van sub-proces + CR
ParamLen    equ               *-Param ; lengte van parameter Param

ends
```

Indien men meer wenst te weten over OS-9, bijvoorbeeld over het aanpassen van OS-9 door het toevoegen en weghalen van evt. zelf gemaakte modules als device-drivers, dan ben ik bereid daar in een evt. volgend verhaal nader op in te gaan. Verder kunnen alle OS-9-gebruikers onder jullie natuurlijk aan mij vragen stellen over OS-9. Daar kunnen we beiden wat van leren. Tot slot wilde ik nog iets vertellen over de verbinding van OS-9 met de buitenwereld.

## Verbinding met andere OS-9 systemen (Atari ST, VME, PC's, UNIX- systemen)

In principe kunnen alle OS-9-systemen onderling verbonden worden via OS-9/NET van Microware zelf. Deze verbinding kan op een aantal manieren tot stand gebracht worden. Via de RS232 is dit voor ieder OS-9- systeem met zo'n poort mogelijk. Zo kunnen Atari's onderling, Atari's en PC's en Atari's en bijvoorbeeld VME-systemen onderling verbonden worden. Verder is het mogelijk via Ethernet hardware OS-9 systemen te koppelen aan bijvoorbeeld SUN-werkstations en andere UNIX-systemen. Hierbij wordt dan gebruik gemaakt van het TCP/IP-protocol. Verder bestaan er voor een aantal systemen (SUN, VAX en PC's) cross-compilers voor C en debuggers etc. om op een werkstation van SUN of op een PC software te kunnen ontwikkelen voor een OS-9-systeem.
Het is dan mogelijk om in te loggen op alle in dit netwerk aanwezige systemen. Verder kan er dan ook gebruik gemaakt worden van andere devices, bijvoorbeeld andere harde schijven, floppydisks. Ook kan men de processor-power van andere systemen benutten en op een ander systeem bijvoorbeeld een reken-intensieve taak starten, omdat in dat systeem een 'floating-point' co-processor aanwezig is.

OS-9/NET is op dezelfde manier opgebouwd als het 'normale' I/O- systeem, dat bestaat uit file-managers, device-drivers and device- descriptors. Verder bevat het nog een aantal utilities om het netwerk te starten, te bekijken etc. Het zogenaamde 'remote inloggen' (het inloggen op een ander systeem dan het jouwe vanaf je eigen systeem) zit niet in OS-9/NET zelf, maar kan als extra utility aangeschaft worden.

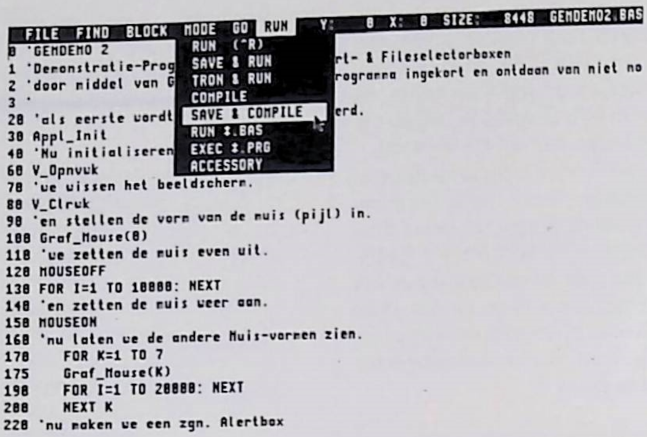
## OS-9's flexibiliteit

OS-9 is mede door deze opzet van modules een erg flexibel besturingssysteem. Het is namelijk mogelijk door het toevoegen of weglaten van modules het besturingssysteem aan te passen aan bijna alle vereisten, bijvoorbeeld bijzondere hardware etc. Verder kan men OS-9 ook draaien zonder schijfeenheden. Alle noodzakelijke modules kunnen zich eenvoudig in ROM bevinden. Beginnend met een systeem zonder schijfeenheden en weinig RAM en eindigend bij een omvangrijk multi-user systeem met diverse terminals, printers, en schijfeenheden is iedere systeem configuratie mogelijk.

**Hans Vermeulen.**

# ATARI heeft voor OMIKRON. gekozen.

# Wanneer kiest U ???




OMIKRON.BASIC V3.0 wordt nu gratis bij elke ST geleverd.

Voor f 24,95 verkrijgbaar bij uw ATARI-dealer.

Inclusief Nederlandstalig handboek.

## 15 redenen waarom U OMIKRON gaat gebruiken:

**1. OMIKRON.Basic 3.0,** een supersnelle interpreter met een nauwkeurigheid van 19 cijfers achter de kommma, gestructureerd programmeren en 99% compatibel met Microsoft-Basic.

**2. OMIKRON.Compiler 3.04,** vertaalt geschreven programma's direkt in echte machinecode (geen P-Code! Sneller dan C).

**3. OMIKRON.Compiler voor de 68881 coprocessor,** voor professioneel rekenkundig gebruik. Werkt met alle rekenkundige coprocessoren.

**4. STATISTIK.LIB** met meer dan 80 toegevoegde funkties, zoals grondstatistiek, grondverdeling, toevalsberekening, variatie-analyse, regressie, correlatie, grafieken.

**5. NUMERIK.LIB** met oplossingen van vergelijkingen en differentiaal-vergelijkingen, numerieke integratie en differentiatie en berekeningen van speciale funkties.

**6. COMPLEX.LIB** voor berekeningen van allerlei funkties, ook met complexe getallen (van niet-analytische, via numerieke integratie tot complexe kromme intergralen).

**7. MIDI.LIB** voor de veeleisende MIDI-gebruiker. Benut nu de MIDI-aansluiting zonder moeilijkheden.

**8. EASYGEM.LIB** laat u in OMIKRON.Basic programmeren onder GEM. Het moeizaam programmeren van dialoogboxen en vensters is hiermee verleden tijd.

**9. FINANZMATHEMATIK.LIB** met rente-, interest- en datum-berekening. U verkrijgt met deze .lib meer dan 400 extra funkties en procedures.

**10. INLINE-ASSEMBLER.LIB** verzorgt het integreren van machinetaal routines binnen OMIKRON.Basic. Ondersteunt alle Motorola-Mnemonics. Werkt met interpreter en compiler.

**11. MASKEN-EDITOR** voor het snel en eenvoudig samenstellen van invoer-vensters en uitvoer-routines.

**12. D.R.A.W.!** Het OMIKRON tekenprogramma met animatie en vele talrijke funkties. Voorzien van een gedocumenteerde source-listing.

**13. ASSEMBLER** vertaalt gemiddeld 1,1 miljoen regels per minuut. Zijn top ligt bij ongeveer 2 miljoen regels. Hiermee is de Assembler veruit de snelste op ST gebied.

**14. ASSEMBLER.MAKRO.** Dit is de grote broer van de vorige. Hiermee wordt het u mogelijk gemaakt om objecten te creëren en met makro-definities te werken.

**15. LINK-IT.** Met Link-it voegt u gecompileerde programma's en Object-Code-Modulen samen. Loopt volledig onder GEM en laat zich door de muis besturen. Assembler-routines worden met een simpel CALL-commando aangeroepen.

## AUTHORISED OMIKRON DEALERS:

**BYTE COMPUTERSHOPS** ZWOLLE, GRONINGEN, LEEUWARDEN, NIJMEGEN / GOTO 66 GOUDA / COMPUTERSHOP UTRECHT / COMMEDIA AMSTERDAM / LOCO DEN HELDER / TELEKODER ROTTERDAM, BERGEN OP ZOOM, BREDA / RADIO MULLER OLDENZAAL

OMIKRON.

IMPORT EN DISTRIBUTIE:

JOTKA COMPUTING - POSTBUS 8183 - 6710 AD EDE

TELEFOON: (08380) 38731 - FAX: (08380) 21675

# Diff Omikron GfA

**Diff is een bekende utility uit de Unix-wereld en wordt gebruikt om verschillen tussen tekstfiles aan te geven. Dit is vooral handig voor programmeurs die meerdere versies van een source bewaren, maar niet meer precies weten wat daartussen de verschillen zijn.**
**Omdat Omikron Basic de nieuwe ST-Basic is, vind ik dat deze best eens aan de hand van een praktijkvoorbeeld vergeleken mag worden met zijn geduchte concurrent van GfA Systemtechnik; de bij dit artikel horende listing is dan ook geschikt voor beide dialecten. De nadruk ligt hierbij echter op GfA 2.02, omdat alleen hiervoor een (onmisbare) compiler beschikbaar is.**
**Diff bevat een aantal subroutines die ook voor andere toepassingen interessant kunnen zijn, zoals een snelle Line Input-routine voor GfA en een Exist() functie voor Omikron.**

## Diff

In de Unix wereld (Unix is een handelsmerk van Bell laboratories) is diff een programma dat twee files (voortaan noemen de linker en de rechter file) regel voor regel met elkaar vergelijkt. Zolang de regels links en rechts overeenkomen geeft diff geen output. De belangrijkste eigenschap van diff zien we als er een regel links met rechts verschilt. Het is dan namelijk heel goed mogelijk dat er op die plaats (bijv. rechts) een regel tussengevoegd is; we willen natuurlijk niet dat diff dan verder alle volgende regels als verschillend beschouwt!
Diff heeft daarom een ingebouwde 'synchronisatie'. Hierbij wordt naar een aantal gelijke opeenvolgende regels gezocht, en wat dan overblijft als verschillend op het scherm aangegeven. Normaal gesproken zie je dan een regelnummer met regelinhoud van de linker gevolgd door regelnummer en regelinhoud van de rechterfile; als een van de regelinhouden 'leeg' is, weet je dat in de andere file op die plaats een regel werd tussengevoegd.

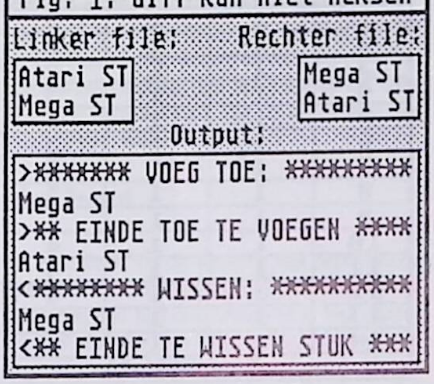
## Heksen

Diff kan natuurlijk niet heksen. Het is niet mogelijk ALLE gelijke regels tussen links en rechts te vinden. Dit zou ten eerste zeer veel tijd kosten (elke regel links moet dan met elke regel rechts vergeleken worden) en levert bovendien niet de gewenste informatie op. Hoe zou diff moeten reageren als een bepaalde regel links 1 keer en rechts meerdere keren voorkomt? Anderszijds zien wij mensen in één oogopslag bepaalde overeenkomsten die diff niet ontdekt (en trouwens niet makkelijk zou kunnen aangeven). In figuur 1 wordt dit aan de hand van twee zeer korte inputfiles weergegeven: diff is niet in staat om op te merken dat de beide regels verwisseld zijn. De output is natuurlijk wel correct, maar bevat meer informatie dan strikt noodzakelijk.

## TOS contra UNIX

Eigenlijk is diff niet de juiste naam voor het programma dat ik geschreven heb; het zou beter 'diffmerge' o.i.d. kunnen heten. In tegenstelling tot de Unix diff geeft dit programma niet alleen maar verschillen als output, maar ook de regels die gelijk zijn. (Zie figuur 1).

Fig. 1: diff kan niet heksen




Onder Unix is het namelijk zeer eenvoudig om de output van diff naar een file te sturen, of zelfs als commando-input voor een editor te laten dienen. TOS kent dit soort faciliteiten niet (denk aan de bugs bij redirection). Als er veel kleine verschillen tussen twee files bestaan, zou de informatie over je scherm vliegen, en zou je alles moeten onthouden.

Maar daar hebben we natuurlijk een computer voor. We sturen de output desgewenst (ook) naar een file. Een ander aspect is natuurlijk dat een regelnummer met een verschil vaak niet veel zegt. Je wilt meestal de omgeving zien waarin dat verschil optreedt om het effect ervan te kunnen begrijpen. Dus geeft 'mijn' diff als output ook de regels die gelijk zijn.

## Syntax

De syntax van diff is als volgt: diff oudefile nieuwefile. In de output geeft diff dan precies aan wat je moet wissen en wat je moet toevoegen om van 'oudefile' 'nieuwefile' te maken.
De output kan direct op het scherm bekeken worden, of later met een editor. Zo heb je een goed overzicht op het effect van de verschillen. Bovendien biedt dit de mogelijkheid om snel een sourcefile in een andere versie te veranderen, nl. gewoon door ofwel de te wissen stukken danwel de toe te voegen stukken in commentaar te veranderen. Dit kan bij grote sources met weinig verschillen tussen meerdere versies een grote besparing aan schijfruimte opleveren!

## Opties en minpunten

Diff biedt een aantal opties die bepaalde verschillen in regels uitschakelen. Zo is het mogelijk spaties en tabs niet mee te beschouwen, zodat een extra spatie of tab niet als een verschil wordt gezien. Ook is het mogelijk om geen onderscheid te maken tussen grote en kleine letters. Deze opties zijn met een beetje programmeerwerk eenvoudig uit te breiden.
Een nadeel van het gebruik van deze opties is dat beide inputfiles twee keer geheel gelezen worden, wat extra tijd kost. Behalve dat diff soms een behoorlijke tijd niets van zich laat horen (zien) bij flink verschillende inputfiles, zijn er nog een paar nadelen. Er is een limiet aan het aantal regels en het aantal bytes dat een inputfile mag bevatten, opgelegd door de hoeveelheid vrij geheugen. Het is wel mogelijk dit te omzeilen, maar dat vond ik voor dit artikel te ver voeren.

## De listing

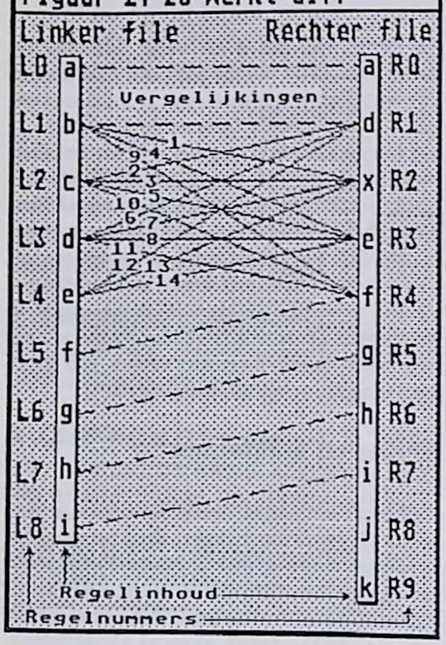
De listing is direct geschikt voor GfA Basic. Er is duidelijk in aangegeven wat moet worden veranderd om hem geschikt te maken voor Omikron. Helaas komt de listing hierdoor een beetje rommelig over.
De regelnummers horen niet bij het programma, maar zijn voor de duidelijkheid aangebracht. Ook in Omikron heb ik deze altijd 'uit' staan.

Zoals gewoonlijk bij dit soort programma's, bestaat het grootste deel van de listing uit I/O routines en initialisatie, terwijl de eigenlijke 'diff' maar uit twee procedures bestaat: Diff en Nog-diff.
Als de listing in GfA wordt ingetypt, is het alleen mogelijk deze in Omikron in te lezen met 'Load block'. Controleer na de aanpassingen met 'Find Error' of je niets vergeten bent. Wordt de listing alleen in GfA gebruikt, dan kunnen natuurlijk alle commentaar regels verwijderd worden.

## Werking Diff

In figuur 2 zijn twee files afgebeeld, waarvan elke regel maar 1 karakter bevat (voor diff maakt dit niets uit, regels zijn regels). Omdat beide regels 0 overeen komen, wordt een van deze regels onveranderd naar de output gestuurd. De regels L1 en R1 komen echter niet overeen.

Figuur 2: zo werkt diff




Nu moet diff zoeken naar een paar gelijke regels, met als criterium dat deze zo dicht mogelijk bij de uitgangspunten in de buurt moet liggen.
Ik heb ervoor gekozen om eerst L1 met R2 en daarna L2 met R1 te vergelijken, gevolgd door L2 en R2. Probeer zelf het patroon dat hierdoor ontstaat in figuur 2 te herkennen. Bij de zesde vergelijking vinden we twee gelijke regels. Daarmee zijn we er echter nog niet: we moeten de synchronisatieproblemen nog voorkomen.

## Synchronisatie

Stel dat in figuur 2 de regels R2 t/m R9 resp. de karakters b,c,d,e,f,g,h en i bevat. Dat zou betekenen dat de twee files gelijk zijn, behalve dat in de rechter file op regel 1 een d is tussengevoegd. Om te bereiken dat diff niet een heleboel verschillen en maar af en toe een enkele gelijke regel meldt, eisen we gewoon dat er vier gelijke regels direct achter elkaar moeten staan.
Dat dit keiharde noodzaak is, is misschien nog beter uit te leggen aan de hand van een GfA Basic listing. Een diff tussen een source en een kopie daarvan, waarin een procedure verplaatst is, zou onverwachte resultaten kunnen opleveren. Dit komt doordat iedere procedure eindigt op Return, met als gevolg dat diff alleen nog maar gelijke Returns vindt, en de rest als verschillend beschouwt. Omdat een Return in de meeste listings gevolgd wordt door een regel met een quote zullen we 3 gelijke regels als minimum moeten stellen; in de listing staat deze default op 4.

## Sub-Diff

In figuur 2 gaat diff dus verder tot en met vergelijking 14. Daar wordt wel aan de eis van 4 opeenvolgende gelijke regels voldaan. Hiermee zijn de inputfiles weer gesynchroniseerd en heeft diff een verschil gevonden: de regels L1 t/m L3 t.o.v. R1 en R2. Omdat zo'n blok nog wel gelijke regels kan bevatten (in dit geval is dat zo) roept diff zichzelf eenmaal recursief aan, uitsluitend voor dat gegeven blok. Nu wordt echter wel naar iedere enkele gelijkheid gekeken, zodat L3 en R1 nu wel als identiek worden aangemerkt. Blijven over als echte verschillen de regels L1, L2 en R2. Li en Ri bevatten altijd de nummers van de eerste verschillende regels. Lm en Rm bevatten bij eerste aanroep van diff (vanuit 'main') de nummers van de laatste regels van de files; bij recursieve aanroep van diff zijn dit de laatste regels van het verschil-blok.
Dat in figuur 2 bij L1 de vergelijkingsnummers als kwadraten oplopen (1,4,9,16,25 etc.) is geen toeval, en geeft al aan wat er gebeurt als in de rechterfile 100 regels werden tussengevoegd; er moeten dan 10000 regels met elkaar vergeleken worden! Dat soort files zijn dus typisch niet voor diff geschikt, of de koffie moet net klaar zijn.

## Nog-Diff

Nog-diff heeft slechts als taak om, uitgaande van twee verschillende regels, te zoeken naar de eerstvolgende gelijke regels, met inachtname van het aantal opeenvolgende regels dat gelijk moet zijn. Geretourneerd worden (in Lj en Rj) de nummers van de laatste regels die nog verschillen, vandaar de naam Nog-diff.
In eerste instantie zal Diff vanuit 'main' aangeroepen zijn. Recur is dan nog false (Fout, 0) en dus zal bij een gevonden verschil Nog-diff aangeroepen worden met 4 voor het aantal gelijke regels. Direct daarna wordt Diff recursief aangeroepen, waarmee alle enkele gelijke regels er nog uit rollen. In tabel 1 en 2 zijn de waarden van de variabelen in Nog-diff te zien, behorende bij figuur 2. Let vooral op het regelmatige patroon waarmee Lj en Rj variëeren.
Door de constructie die wordt uitgevoerd als Lj=Ljmax en Rj=Rjmax wordt voorkomen dat Nog-diff voorbij het einde van een van de files (of van het gegeven sub-blok bij recursie van diff) probeert te lezen. In tabel 2 is te zien dat daarom vergelijking 4 en 5 overgeslagen worden; Reob = true (Rechts End Of Block).

## Werking Diff globaal

Het programma vraagt de gebruiker na opstarten direct om de twee inputfiles. Daarna dient gekozen te worden voor het type uitvoer: naar het scherm (een 'More' is ingebouwd), naar een file of beide. Vervolgens worden de inputfiles elk regelsgewijs in een stringarray gelezen, waarna de gebruiker om eventuele opties gevraagd wordt. Bijv. de optie 'Spaties negeren' zal dan uit beide stringarrays alle spaties verwijderen m.b.v. de procedure Verwijder.
De inputfiles, die ondertussen weer gesloten zijn, worden opnieuw geopend, waarna Diff wordt aangeroepen. Deze vergelijkt de regels uit de stringarrays, maar de regels die naar output geschreven worden komen vanaf disk (de stringarrays zijn immers gemuteerd door opties).
Het ligt voor de hand om het programma gecompileerd te gebruiken. Zorg er bij de GfA-compiler voor dat het programma minstens in 'loops' onderbroken kan worden, voor het geval dat de inputfiles lang zijn en erg

Tabel 1: Nog_diff met Li=1 Ri=1 Lm=8 Rm=9 N=4

| Verg. | Ljmin | Rjmin | Ljmax | Rjmax | Lj | Rj | L$ | R$ | Leob | Reob |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 2 | 1 | 2 | b | x | Fout | Fout |
| 2 | 1 | 1 | 2 | 2 | 2 | 1 | c | d | Fout | Fout |
| 3 | 1 | 1 | 2 | 2 | 2 | 2 | c | x | Fout | Fout |
| 4 | 1 | 1 | 3 | 3 | 1 | 3 | b | e | Fout | Fout |
| 5 | 1 | 1 | 3 | 3 | 2 | 3 | c | e | Fout | Fout |
| 6 | 1 | 1 | 3 | 3 | 3 | 1 | d | d | Fout | Fout |
| 7 | 1 | 1 | 3 | 3 | 3 | 2 | d | x | Fout | Fout |
| 8 | 1 | 1 | 3 | 3 | 3 | 3 | d | e | Fout | Fout |
| 9 | 1 | 1 | 4 | 4 | 1 | 4 | b | f | Fout | Fout |
| 10 | 1 | 1 | 4 | 4 | 2 | 4 | c | f | Fout | Fout |
| 11 | 1 | 1 | 4 | 4 | 3 | 4 | d | f | Fout | Fout |
| 12 | 1 | 1 | 4 | 4 | 4 | 1 | e | d | Fout | Fout |
| 13 | 1 | 1 | 4 | 4 | 4 | 2 | e | x | Fout | Fout |
| 14 | 1 | 1 | 4 | 4 | 4 | 3 | e | e | Fout | Fout |
| Teruggave Nog_diff: | | | | | 3 | 2 | | | | |

Tabel 2: Nog_diff met Li=1 Ri=1 Lm=3 Rm=2 N=1

| Verg. | Ljmin | Rjmin | Ljmax | Rjmax | Lj | Rj | L$ | R$ | Leob | Reob |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 2 | 2 | 1 | 2 | b | x | Fout | Fout |
| 2 | 1 | 1 | 2 | 2 | 2 | 1 | c | d | Fout | Fout |
| 3 | 1 | 1 | 2 | 2 | 2 | 2 | c | x | Fout | Fout |
| 6 | 3 | 1 | 3 | 2 | 3 | 1 | d | d | Fout | Waar |
| Teruggave Nog_diff: | | | | | 2 | 0 | | | | |

verschillen. Er bestaat qua werking een klein verschil tussen de beide Basic- versies; zie bij EOF en LINE INPUT.

## Uitvoeringstijden

In tabel 3 zijn de uitvoeringstijden van diff te zien. Het betreft in alle gevallen een aangepaste versie van diff, die geen gebruikers-handelingen vraagt. Als eerste en laatste statement wordt de tijd opgenomen. Er wordt alleen naar een file geschreven, dus de scherm uitvoer telt niet mee. Als opties werden S,T en H meegegeven.

| Tabel 3: tijden (seconden) | |
|---|---|
| GfA 2.02 Interpreter | 30.1 |
| Omikron 3.0 Interpreter | 23.0 |
| GfA 3.04 Interpreter | 22.1 |
| GfA 2.02 Compiler | 13.2 |
| Omikron 3.04 Compiler | 12.1 |

De tijd staat dus voor twee maal twee files lezen, opties en vervolgens diff uitvoeren, en een file schrijven. De invoerfiles waren beide 12874 bytes groot (een versie van diff) met daarin twee t.o.v. elkaar verschoven procedures, en stonden op ramdisk. Ook de uitvoerfile werd naar ramdisk geschreven, om zoveel mogelijk externe factoren te elimineren. Bij de GfA-compiler was de optie 'Break in Loops' aangezet.

## Omikron contra GfA

De ontwikkeling van diff ben ik in GfA begonnen, en ben halverwege op Omikron overgestapt. Het laatste stuk is weer m.b.v. GfA geschreven. Dit geeft hetzelfde gevoel als vandaag in Engeland en morgen in Nederland autorijden: het is niet handig, maar alles went.
Toch zit niet alleen alles onder andere toetsen; er bestaan grotere verschillen tussen de beide Basic's dan ik had verwacht. De verschillen die ik tegengekomen ben, zal ik nu behandelen. Helaas vond ik ook een vervelende bug in Omikron Basic V3.0: na het gelijktijdig indrukken van ALT-HELP (screendump) zit de ST muurvast (ook nog na 30 seconden). Met een druk op de resetknop wordt het scherm zwart, en blijft dat ook. Helaas was ik daarmee dus ook m'n ramdisk kwijt; uit/aan schakelen van de ST bleek onvermijdelijk.

## Muis

GfA is een stuk muis-vriendelijker dan Omikron. Bij Omikron staat default de muispijl uit, met als gevolg dat je bij een evt. AES alertbox 'in het donker' naar de gewenste button moet zoeken. Zet je de muis aan, dan krijg je een heleboel pijltjes bij alle print commando's, vooral als daarbij het scherm scrollt. GfA vangt dit meestal goed op door intern vóór outputstatements de muis uit te schakelen. Ik heb echter ook bij GfA wel eens met een alert zonder muispointer gezeten bij een gecompileerd programma waarin echt geen Hidem te vinden was.
In diff staat normaal gesproken de muis aan; deze wordt alleen bij schermuitvoer uitgeschakeld.

## Fileselect

Niet alleen accesoires laten grijze vlakken achter; in Omikron doet de fileselector dit ook. In GfA gebeurt hetzelfde, maar deze slaat juist voor de fileselect aanroep het hele scherm op, en herstelt dit na afloop weer. In het programma wordt dit door de procedure Fsel-screen (alleen voor Omikron) verzorgd. Houd er bij eigen gebruik van deze routine rekening mee dat deze tijdelijk 32000 bytes nodig heeft!
Helaas zit er nog een bug in Omikron V3.0. Bij aanroep van fileselector met een 'preselectie' - dat is de naam die rechtsboven in de fileselector al is ingevuld - wordt deze naam ingekort tot 8 karakters. Dit komt, omdat er intern voor de aanroep van de AES fileselector te weinig stackruimte gegeven wordt. De AES saved de processor-registers op de stack, en juist deze stack groeit naar de gegeven preselectstring toe. Zo worden de laatste 4 van 12 voor deze string gereserveerde bytes met de inhoud van een adresregister overschreven. Een geluk bij ongeluk is hierbij dat op een 68000 machine de 'linker' byte van een adres meestal 0 is, zodat dit als de string-terminator wordt gezien! Korte of geen preselects meegeven, dus.

## EOF

GfA en Omikron houden er verschillende interpretaties op na wat EOF() betekent. Ten eerste schrijft Omikron altijd een EOF-marker (ASCII 26, CTRL-Z) bij Close van een file die voor schrijven was geopend, ook al is er verder niets naar die file geschreven. GfA schrijft nooit zo'n EOF marker.
Ook bij lezen zijn er belangrijke verschillen. Stel we hebben een file van 100 bytes, waarvan de laatste een EOF is. Omikron zal dan EOF()=true geven als 99 bytes gelezen zijn, GfA als er 100 gelezen zijn. In Omikron levert A$=INPUT$(100, 1) bij die file een 'Read past end of file' op, GfA leest ze wel in.
Gelukkig accepteert Omikron ook files die niet op EOF eindigen; in het bovenstaande voorbeeld is het resultaat dan hetzelfde als bij GfA. Voor diff betekent e.e.a. dat de outputfiles qua lengte 1 byte zullen verschillen.

## Line Input

In beide Basic's is de werking van deze functie hetzelfde. Een belangrijk verschil zit echter in de uitvoeringstijd hiervan. Omikron is hier duidelijk veel slimmer dan GfA: het inlezen van een source van diff (13 Kbytes, vanaf ramdisk) met Line Input statements duurt in Omikron 1,1 seconde, in GfA 3.04 9,2 seconden en in GfA 2.02 maar liefst 17,1 seconden! (Alles interpreter tijden). Dit verschil onstaat doordat Omikron intern buffert, d.w.z. blokken van een file 'vooruit leest'. GfA daarentegen leest byte voor byte m.b.v. GEMDOS-functies, en juist deze laatste functies zijn zo traag. Hoewel GEMDOS intern ook buffert, is deze toch relatief langzaam omdat voor iedere te lezen byte een Mediachange call uitgevoerd wordt; sinds de vorige byte kan er van disk gewisseld zijn. Bij elkaar kost dat allemaal veel tijd.

## Lees-Regel

Als alternatief voor GfA heb ik daarom in diff een bufferroutine opgenomen, die bovenstaande file in 2,2 seconden inleest, en daarbij ook nog flexibeler is dan de standaard Line Input. Leesregel staat namelijk toe dat de laatste regel van een file niet met CRLF (Carriage Return Line Feed, $0D0A) is afgesloten. De optimale buffergrootte blijkt voor GfA 512 bytes te zijn. Dezelfde routine in Omikron is het snelst bij 128 bytes, en doet er dan 3,2 seconden over! Bovendien treedt het EOF-probleem op als de inputfile eindigt op Chr$(26). Deze routines dus alleen voor GfA toepassen!
Lees-regel zoekt naar de eerste LF in de buffer. Wordt deze gevonden, dan zal de string tot 1 byte voor de LF teruggegeven worden in Regel$, waarna de string inclusief CRLF uit de buffer wordt verwijderd. Wordt geen LF gevonden, dan zal de buffer aangevuld worden met een blok uit de file ter groote van Blk-size of, indien minder, het nog t/m het einde van de file resterende aantal bytes. Als de file geheel gelezen is, en in de buffer komt geen LF meer voor, dan wordt de gehele buffer in Regel$ alsmede EOF=True teruggegeven.

## Exist

Een handige GfA-functie die ik in Omikron mis, is Exist(filenaam). Deze functie gebruik ik altijd voor de aanroep van de fileselector om te zorgen dat ik een actuele directory te zien krijg, ook na een diskwissel. Exist probeert de gegeven file te openen voor lezen met de GEMDOS Fopen call. Als de file bestaat, krijg je een positief getal terug. In dat geval moet de file weer worden gesloten met Fclose. Krijg je een negatief getal terug, dan bestond de file niet, en hoeft dan natuurlijk ook niet meer gesloten te worden. Waarom de Exist functie fileselector problemen voorkomt, weet ik niet precies. Kwaad kan het niet, en het help echt. Verder wordt Exist ook gebruikt waar hij voor bedoeld is.

## Geheugengebruik

Omikron en GfA zijn beide nogal kwistig met geheugen bij het gebruik van stringarrays. Een lege string uit zo'n array kost 6 bytes bij beide merken. Is de string niet leeg, dan komen er in GfA 4 bytes bij, terwijl er in Omikron zelfs 10 bij komen! Omdat we bij diff van te voren niet weten hoeveel regels een inputfile bevat, gaan we voor het gemak uit van gemiddeld 12 karakters per regel, resulterend in 10 bytes per string (de CRLF wordt niet opgeslagen). In de praktijk blijkt dit een redelijke waarde. Dus zullen we bij Omikron gemiddeld op 26 bytes (totaal) per string moeten rekenen; bij GfA op 20. Als we grotere files aan willen kunnen, is het natuurlijk mogelijk het in het artikel 'Snelle Tekstuitvoer' (ST 16) toegepaste principe van fileopslag te gebruiken.

## Merkwaardigheden

Nog een paar zaken waar ik tegenaan gelopen ben. Inkey$ levert in Omikron een string terug van 4 bytes! Zie handboek; je moet het maar weten.

Omikron is wat inconsequent met het gebruik van het tekentje (matje of hekje) voor filehandles. Dit mag niet gebruikt worden bij OPEN, CLOSE, LOF, en EOF maar moet bijv. gebruikt worden bij LINE INPUT 1,A$.

Omikron zoekt altijd op A: (ik heb geen harddisk, en werk altijd vanaf ramdisk D:) naar een OM-BASIC.INF. Waarom niet in de directory waar OM-BASIC.PRG gestart werd? Ook 'Save setings' gaan ongewild naar A:.

Let er op dat variabelen zonder postfix in Omikron long integers zijn, hetzelfde als Var% in GfA.

Gebruik in GfA NOOIT de volgende uitdrukking: New!=true. Opvallend is dat de editor de 't' van true niet in een hoofdletter verandert. Maar veel erger is het effect van deze toekenning in een groot, nog niet gesaved programma!

## Overeenkomsten

Een belangrijk verschil met andere programmeertalen, maar op dezelfde wijze in beide Basic's geïmplementeerd is het gebruik van lokale variabelen. In tegenstelling tot bijv. Pascal zijn deze volgens een boomstructuur opgezet. Stel, in 'main' krijgt een variable A de waarde 0. In een procedure X, waarin A als lokaal is gedeclareerd, krijgt A de waarde 1. De aanroep van een procedure Y (waarin uitsluitend Print A staat) vanuit X zal een 1 op het scherm schrijven. Natuurlijk zal een aanroep van Y vanuit main een 0 printen. Het principe is dus, dat na een declaratie van een lokale variabele A in procedure X deze als globaal gezien wordt voor alle sub-procedures van X.

Intern is dit erg eenvoudig. Het betekent gewoon dat bij het Local statement de oude waarde van de variabele op de stack gezet wordt, om bij Return weer te worden hersteld.

Nogmaals een waarschuwing voor For Next. Dit is eigenlijk een Repeat Until met ingebouwde teller. Dit betekent dat de lus ALTIJD minstens 1 keer wordt doorlopen, bijv. bij For A=1 To 0 ... Next A.

## Gebruiksgemak

Hoewel je aan de Omikron editor moet wennen, is deze zeer snel, en biedt zeker meer mogelijkheden dan GfA 2.02. Bijvoorbeeld de Query replace werkt handiger dan de CTRL-R die je in GfA nodig hebt.

Jammer vind ik dat het outputscherm van Omikron verdwenen is na even in de editor gekeken te hebben. Bij GfA maak ik vaak gebruik van deze mogelijkheid. Ook onnodig is het doorschieten ver voorbij het einde van de source met CTRL-cursor down in Omikron.

Een nadeel van GfA is dat het niet mogelijk is om hiermee een ASCII file uit 1ST-WORD in te lezen. Dit komt doordat dit soort files afgesloten worden met een extra CRLF. (Weer een andere interpretatie van hoe een end-of-file er uit moet zien!) GfA geeft dan de fantastische foutmelding: Not an ASCII-file. Je kunt dit verhelpen door in 1ST-WORD de laatste CR te verwijderen, zodat de cursor vanaf de laatste tekstregel niet verder naar beneden kan.

Vanuit Omikron is het niet mogelijk een ASCII-file te saven die niet ineens barst van de %L postfixes. Belangrijke voordelen van Omikron t.o.v. GfA 2.02 zijn het gebruik van var-parameters (R Var) en meerregelige functies. Leuk van Omikron is dat je met een reset (ja, dat knopje achterop) meestal weer in de interpreter terugkomt. Handig bij B52 programma's.

## Misverstand

Door een foutje zijn er op de Omikron Basic V3.0 disk een aantal files terechtgekomen die er niet horen. Het betreft hier tekstfiles die door Terminal Software voor eerdere versies von Omikron Basic zijn gemaakt. Het Omikron introductie pakket bevatte destijds o.a. een enkelzijdige, maar met tien sectoren per track geformatteerde interpreter-disk. Omdat een back-up hiervan via diskcopy niet werkt, schreef ik Terminal een briefje met onder meer die opmerking. Terminal heeft die opmerkingen in een tekstfile op de interpreter disk opgenomen. Helaas is deze tekstfile 'LEES-DIT.1ST' ook op de V3.0 disk terecht gekomen. Deze disk is echter standaard dubbelzijdig geformatteerd (jammer voor mensen met een enkelzijdige drive). Verder staan in deze en andere tekstfiles nog meer foutjes. Bovendien wordt overal nog naar Terminal Software verwezen. Dit bedrijf is failliet, en alle Omikron activiteiten zijn overgenomen door Atari Benelux B.V. en Jotka Computing.

## Conclusie

Met dit artikel heb ik eens een andere visie op de twee bekendste Basic's willen geven dan in de meeste recensies te vinden is. Natuurlijk geeft dit maar een beperkt beeld van de geboden faciliteiten. Voor dit type applicatie (geen gebruik van GEM etc.) maakt het naar mijn mening niet veel uit welke Basic er gebruikt wordt, hoewel ik toch een lichte voorkeur heb voor GfA. Een ergerlijk nadeel van GfA vind ik dat de nieuwe compiler nog niet beschikbaar is, terwijl de Omikron-compiler V3.04 al in de winkels ligt. Ook zal voor veel mensen de prijs van de Omikron-interpreter relatief gunstig zijn (f24,95); dat de GfA-compiler voor interpreter bezitters (bijna) niets zal kosten schept niet veel vertrouwen in de werkdrang bij GfA Sytemtechnik. Het kon wel eens heel lang wachten worden, op die compiler.

## P.S.

Bezitters van oudere Omikron-compilers kunnen deze bij Jotka Computing, Postbus 8183, 6710 AD Ede, laten updaten. Een update vanaf 3.0 naar de nieuwste versie kost fl 17,50; voor oudere versies moet fl 49,- betaald worden (prijzen per 14-3-'89).

Erik van Straten.

*Literatuur:*


*Handleiding Omikron Basic*

*Handleiding GfA Basic*

*Baertels, Egel, Merine, Schneider: Das GfA Basic Buch (Heim Verlag)*

*Jankowski, Reschke, Rabich: Atari ST Profibuch*

*Brueckmann, Englisch, Gerits: Atari ST Intern*

*SunOS (TM) Manual Pages*

*V. Straten: Snelle tekstuitvoer ST16*

*TOS: Trauma's, Tips & Trucs (deel 3) ST15*


```
001 ' ••••••••••••••• LISTING 1 ••••••••••••••
002 Cpr$="DIFMERGE.BAS diff utility (PD) "
003 Cpr$=Cpr$+"V1.19 (C) 30-3-'89 "
004 Cpr$=Cpr$+"door Erik van Straten."
005 '
006 ' Aanpassen voor Omikron:
007 ' Vervang alle (Replace all):
008 '     {Procedure}      door {DEF PROC}
009 '     {%}              door {}
010 '     {!}              door {'}
011 '     {Open "I",#}     door {OPEN "I",}
012 '     {Open "O",#}     door {OPEN "O",}
013 '     {<RETURN>",}     door {<RETURN>";}
014 '     {' ==>}          door {}
015 '  Verwijder laatste 3 procedures
016 '  Verwijder alle {@} behalve in
017 '    procedure Meer bij PRINT @(...
018 '  Verwijder alle regels met {'<==}
019 '  Vervang alle regels met
020 '  {Exit If ...} door {IF ... THEN EXIT}
021 '
022 ' Belangrijke GLOBALE variabelen:
023 '       (Denk voor GfA de % erbij)
024 ' L            huidige schermregel in Meer
025 ' N,N2         min. aantal gelijke regels
026 ' Maxr         max. aantal regels per file
027 ' Maxb      max. aantal bytes voor 2 files
028 ' L$(),R$() fileregels, evt. aangepast
029 '              door opties
030 ' Op$          bevat opties voor diff
031 ' Lf$,Rf$,Of$ filenamen
032 ' Li,Ri        regeltellers binnen Diff
033 ' Lj,Rj        Eigenlijk Local binnen Diff
034 '              en Nodiff: de nummers van
035 '              laatste verschillende regels
036 ' Lm,Rm        laatste regelnummer per file
037 '              of (bij recursie) van blok
038 ' Lregel$,Rregel$ uit file gelezen regel
039 ' F$,Select$,Pathfile$,Outf$  gebruikt
040 '              voor fileselect
041 ' Fout,Waar Booleans voor True en False
042 ' (eigenlijk alleen nodig voor Omikron)
043 '
044 ' Alleen voor GfA (in Lees_?regel)
045 ' Blk_size    aantal bytes voor Input$
046 ' Lfp,Rfp     file pointers (in bytes)
047 ' Leof,Reof   end of file gesignaleerd
048 ' Lfl,Rfl     file lengte (in bytes)
049 ' Lb$,Rb$     blok buffers
050 '
051 N%=4  !  Min. aantal regels voor Match
052 '                        neem deze 3 of 4
053 N2%=1 !  Aantal regels voor sub-match:
054 '                        neem deze 1 of 2
055 Pathfile$="\*.*"
056 Uitf$="OUT.TXT"
057 '
058 ' ==>MOUSEON ! muis aan tenzij print
059 Fout%=0
060 Waar%=-1
061 ' ==>Dummy=FN Exist(Pathfile$)
062 Dummy%=Exist(Pathfile$) !<==
063 ' ==>Maxr=(FRE("")-8000)/52
064 Maxr%=(Fre(0)-8000)/40 !<==
065 Maxb%=Maxr%*20
066 Dim R$(Maxr%)
067 Dim L$(Maxr%)
068 '
069 @Info
070 @Haal_file_namen
071 Repeat
072   L%=0 ! schermregel teller
073   Open "I",#1,Lf$
074   Open "I",#2,Rf$
075   @Zet_buf_globals !<==
076   ' ==>Ok=(LOF(1)+LOF(2))<=Maxb
077   Ok%=(Lfl%+Rfl%)<=Maxb% !<==
078   If Ok% Then
079     @Haal_input_files
080     ' ==>Ok=EOF(1) And EOF(2)
081     Ok%=Leof% And Reof% !<==
082     Close
083   Endif
084   If Not Ok% Then
085     A$="Files te groot en/of|"
086     A$=A$+"teveel regels."
087     ' ==>A$="[1]["+A$+"][SORRY]"
088     ' ==>FORM_ALERT(1,A$,A)
089     Alert 1,A$,1,"SORRY",A%  !<==
090   Else
091     @Doe_diff
092   Endif
093   For I%=0 To Lm%
```

```
094     L$(I%)=""
095   Next I%
096   For I%=0 To Rm%
097     R$(I%)=""
098   Next I%
099   ' ==>MOUSEOFF
100   Repeat
101   Until Inkey$=""
102   Input "Klaar. <RETURN>",A$
103   Cls
104   ' ==>MOUSEON
105   @Info
106   @Haal_file_namen
107 Until Fout%=Waar%
108 '
109 Procedure Info
110   ' ==>MOUSEOFF
111   Cls
112   ' ==>PRINT Chr$(27);"f";' Cursor uit
113   Print Cpr$
114   Print "Beschikbaar geheugen: 2 maal ";
115   Print Maxb%/2;" bytes";
116   Print " met ";Maxr%;" regels."
117   ' ==>MOUSEON
118 Return
119 '
120 Procedure Doe_diff
121   Open "I",#1,Lf$
122   Open "I",#2,Rf$
123   @Zet_buf_globals !<==
124   If Uit_type%>1 Then
125     Open "O",#3,Of$
126   Endif
127   @Opties
128   A$="*** Gegevens om van "+Lf$
129   A$=A$+" de file "+Rf$+" te maken."
130   @Pr(A$)
131   @Pr("*** Opties: "+Op$)
132   If Uit_type%=2 Then ! alleen file
133     ' ==>MOUSEOFF
134     Print "File wordt geschreven..."
135     ' ==>MOUSEON
136   Endif
137   @Diff(0,Lm%,0,Rm%,Fout%)
138   Close
139 Return
140 '
141 Procedure Diff(Li%,Lm%,Ri%,Rm%,Recur%)
142   Local Lj%,Rj%
143   While (Li%<=Lm%) And (Ri%<=Rm%)
144     If L$(Li%)=R$(Ri%) Then
145       @Uitvoer_gelijk(Li%,Ri%)
146       Li%=Li%+1
147       Ri%=Ri%+1
148     Else
149       If Recur% Then
150         @[illegible]_diff(Li%,Lm%,Ri%,Rm%,N2%)
151         @Uitvoer_diff(Li%,Lj%,Ri%,Rj%)
152       Else
153         @[illegible]_diff(Li%,Lm%,Ri%,Rm%,N%)
154         @[illegible](Li%,Lj%,Ri%,Rj%,Waar%)
155       Endif
156       Li%=[illegible]+1
157       Ri%=[illegible]+1
158     Endif
159   Wend
160   @Uitvoer_diff(Li%,Lm%,Ri%,Rm%)
161 Return
162 '
163 Procedure [illegible]_diff(Li%,Lm%,Ri%,Rm%,N%)
164   Local Ljmin%,Ljmax%,Rjmin%,Rjmax%
165   Local Leob%,Reob%,Match%
166   Lj%=Li%          !   lopende tellers
167   Rj%=Ri%+1        !   let op, deze 1 hoger
168   Ljmin%=Li%       !   minima
169   Rjmin%=Ri%
170   Ljmax%=Li%+1     !   maxima
171   Rjmax%=Ri%+1
172   Match%=Fout%
173   Leob%=Fout%      !   einde van blok
174   Reob%=Fout%
175   While Lj%<=Lm% And Rj%<=Rm%
176 '
177     Match%=L$(Lj%)=R$(Rj%)
178     I%=1
179     While (Match%<>0) And (I%<N%)
180       Exit If (Lj%+I%)>Lm%
181       Exit If (Rj%+I%)>Rm%
182       Match%=L$(Lj%+I%)=R$(Rj%+I%)
183       I%=I%+1
184     Wend
185     Exit If Match%
186 '
187     If (Lj%<Ljmax%) Or (Rj%<Rjmax%) Then
188       If Lj%<Ljmax% Then
189         Lj%=Lj%+1
190         If Lj%=Ljmax% Then
191           Rj%=Rjmin%
192         Endif
193       Else
194         Rj%=Rj%+1
195       Endif
196     Else !        Lj=Ljmax And Rj=Rjmax
197       Ljmax%=Ljmax%+1
198       If Ljmax%>Lm% Then
199         Leob%=Waar%
200         Ljmax%=Lm%
201         Rjmin%=Rj%+1
202       Endif
203       Rjmax%=Rjmax%+1
204       If Rjmax%>Rm% Then
205         Reob%=Waar%
206         Rjmax%=Rm%
207         Ljmin%=Lj%+1
208       Endif
209       Rj%=Rj%+1
210       If Rj%>Rm% Then
211         Rj%=Rjmin%
212       Endif
213       Lj%=Ljmin%
214     Endif
215     Exit If Leob% And Reob%
216   Wend
217   Rj%=Rj%-1 ! Geef de nummers van de
218   Lj%=Lj%-1 ! laatste verschil regels
219 Return ! Lj, Rj
220 '
221 Procedure Uitvoer_diff(Li%,Lj%,Ri%,Rj%)
222   Local I%
223   If Lj%>=Li% Then
224     @Prx("<******** WISSEN: **********")
225     I%=Li%
226     While I%<=Lj%
227       ' ==>LINE INPUT #1,Lregel$
228       @Lees_lregel !<==
229       @Prx(Lregel$)
230       I%=I%+1
231     Wend
232     @Prx("<** EINDE TE WISSEN STUK ***")
233   Endif
234   If Rj%>=Ri% Then
235     @Prx(">******* VOEG TOE: **********")
236     I%=Ri%
237     While I%<=Rj%
238       ' ==>LINE INPUT #2,Rregel$
239       @Lees_rregel !<==
240       @Prx(Rregel$)
241       I%=I%+1
242     Wend
243     @Prx(">** EINDE TOE TE VOEGEN ****")
244   Endif
245 Return
246 '
247 Procedure Uitvoer_gelijk(Li%,Ri%)
248   ' ==>LINE INPUT #1,Lregel$
249   @Lees_lregel !<==
250   ' ==>LINE INPUT #2,Rregel$
251   @Lees_rregel !<==
252   @Pr(Lregel$)
253 Return
254 '
255 Procedure Opties
256   Local I%
257   ' ==>MOUSEOFF
258   Print
259   Print "Maak een keuze uit de ";
260   Print "volgende opties:"
261   Print
262   Print "(S)paties negeren"
263   Print "(T)abs negeren"
264   Print "(H)oofd/kleine letters negeren"
265   Print
266   Print "(Opties kosten vaak veel tijd)"
267   Repeat
268   Until Inkey$=""
269   Input "Keuzes: S,T,H <RETURN>",Op$
270   If Op$<>"" Then
271     Print "Moment AUB..."
272   Endif
273   Print
274   ' ==>MOUSEON
275   Op$=Upper$(Op$)
276   If Instr(Op$,"S") Then
277     @Verwijder(" ")
278   Endif
279   If Instr(Op$,"T") Then
280     @Verwijder(Chr$(15))
281   Endif
282   If Instr(Op$,"H") Then
283     For I%=0 To Lm%
284       L$(I%)=Upper$(L$(I%))
285     Next I%
286     For I%=0 To Rm%
287       R$(I%)=Upper$(R$(I%))
288     Next I%
289   Endif
290 Return ! Op$
291 '
292 Procedure Verwijder(Uit$)
293   Local P%,T$,X%,I%
294   For I%=0 To Lm%
295     T$=L$(I%)
296     P%=Instr(T$,Uit$)
297     While P%>0
298       X%=Len(T$)-P%
299       T$=Left$(T$,P%-1)+Right$(T$,X%)
300       P%=Instr(T$,Uit$)
301     Wend
302     L$(I%)=T$
303   Next I%
304   For I%=0 To Rm%
305     T$=R$(I%)
306     P%=Instr(T$,Uit$)
307     While P%>0
308       X%=Len(T$)-P%
309       T$=Left$(T$,P%-1)+Right$(T$,X%)
310       P%=Instr(T$,Uit$)
311     Wend
312     R$(I%)=T$
313   Next I%
314 Return
315 '
316 Procedure Haal_input_files
317   @Prnt("Files worden gelezen...",-1)
318   Lm%=-1
319   ' ==>WHILE (Lm<Maxr) AND NOT EOF(1)
320   While (Lm%<Maxr%) And Not Leof% !<==
321     Lm%=Lm%+1
322     ' ==>LINE INPUT #1,L$(Lm)
323     @Lees_lregel    !<==
324     L$(Lm%)=Lregel$ !<==
325   Wend
326   If L$(Lm%)="" Or L$(Lm%)=Chr$(26) !<==
327     Dec Lm% !laatste regel=leeg/EOF !<==
328   Endif                              !<==
329   Rm%=-1
330   ' ==>WHILE (Rm<Maxr) AND NOT EOF(2)
331   While (Rm%<Maxr%) And Not Reof% !<==
332     Rm%=Rm%+1
333     ' ==>LINE INPUT #2,R$(Rm)
334     @Lees_rregel    !<==
335     R$(Rm%)=Rregel$ !<==
336   Wend
337   If R$(Rm%)="" Or R$(Rm%)=Chr$(26) !<==
338     Dec Rm%                          !<==
339   Endif                              !<==
340 Return ! L$(), R$(), Lm, Rm
341 '
342 Procedure Haal_file_namen
343   Local R%,F$,Lezen%,A$
344   Lezen%=Waar% !  Open file voor lezen
345   @Prnt("Input file 1: ",0)
346   @Fsel(Pathfile$,"",Lezen%)
347   Lf$=F$
348   @Prnt("Input file 2: ",0)
349   @Fsel(Pathfile$,"",Lezen%)
350   Rf$=F$
351   A$="Uitvoer naar:           "
352   A2$="SCHERM|FILE|BEIDE"
353   ' ==>A$="[2]["+A$+"]["+A2$+"]"
354   ' ==>@FORM_ALERT(3,A$,Uit_type)
355   Alert 2,A$,3,A2$,Uit_type%    !<==
356   If Uit_type%>1 Then
357     @Prnt("Output file : ",0)
358     @Fsel(Pathfile$,Uitf$,Not Lezen%)
359     Of$=F$
360   Endif
361 Return ! Lf$, Rf$, Uit_type, Of$
362 '
363 Procedure Fsel(Pf$,Select$,Lezen%)
364   Local A%,A$,Ok%
365   Repeat
366     ' ==>F$=Select$
367     ' ==>Fsel_Screen(Pf$,F$,F)
368     Fileselect Pf$,Select$,F$ !<==
369     @Prnt(F$,-1)
370     ' ==>IF F=Fout THEN
371     If Right$(F$)="\" Or F$="" !<==
372       @Einde
373     Endif
374     ' ==>Ok=FN Exist(F$)
375     Ok%=Exist(F$) !<==
376     If Not Lezen% Then ! file schrijven
377       If Ok% Then !        file bestaat al
378         If F$=Lf$ Or F$=Rf$ Then
379           A$="Ik kan geen input-file"
380           A$=A$+"|overschrijven."
381           ' ==>A$="[1]["+A$+"][SORRY]"
382           ' ==>@FORM_ALERT(1,A$,A)
383           Alert 1,A$,1,"SORRY",A%      !<==
384           Ok%=Fout%
385         Else
386           A$="Output file bestaat al.|"
387           A$=A$+"Overschrijven?"
388           ' ==>A$="[1]["+A$+"][NEE|JA]"
389           ' ==>@FORM_ALERT(2,A$,A)
390           Alert 1,A$,2,"NEE|JA",A%  !<==
391           Ok%=(A%=2)
392         Endif
393       Else
394         Ok%=Waar% !nieuwe file schrijven
395       Endif
396     Endif
397   Until Ok%=Waar%
398 Return ! F$
399 '
400 Procedure Prx(T$) ! print string inverse
401   Print Chr$(27);Chr$(112);
402   @Pr(T$) ! print naar file + scherm
403   Print Chr$(27);Chr$(113);
404 Return
405 '
406 Procedure Pr(T$)
407   If Uit_type%=2 Then       ! alleen file
408     Print #3,T$
409   Else
410     If Uit_type%=1 Then  ! alleen scherm
411       @Prnt(T$,-1)
412       @Meer
413     Else                              ! beide
414       Print #3,T$
415       @Prnt(T$,-1)
416       @Meer
417     Endif
418   Endif
419 Return
420 '
421 Procedure Prnt(T$,Cr%)
422   ' ==>MOUSEOFF
423   If Cr% Then
424     Print T$
425   Else
426     Print T$;
427   Endif
428   ' ==>MOUSEON
429 Return
430 '
431 Procedure Meer
432   Local R$
433   L%=L%+1
434   If L%>23 Then
435     @Prnt("<Meer>",0)
436     Repeat
437     Until Inkey$=""
438     Repeat
439       R$=Inkey$
440     Until R$<>""
441     ' ==>MOUSEOFF:PRINT @(24,0); SPC(6);
442     ' ==>PRINT @(24,0);:MOUSEON
443     Print At(1,25);Spc(6); !<==
444     Print At(1,25); !<==
```

*(vervolg op pag. 53)*

# G + PLUS

## Codehead Software maakt wat Atari niet is gelukt.

**Na de vele artikelen over GDOS zult u nu wel weten wat GDOS is. GDOS maakt het mogelijk om met verschillende karaktersets (fonts) te werken en heeft een standaard uitvoer, onafhankelijk van het uitvoerapparaat. Hierdoor zijn WYSIWYG (What-You-See-Is-What-You-Get) programma's te ontwikkelen. Programma's die van GDOS gebruik maken zijn er inmiddels in overvloed. Tekstverwerkers, tekenprogramma's, desktop-publishing programma's en spreadsheets (b.v. LDW-Power) maken gebruik van GDOS om met verschillende karaktersets te kunnen werken en om een nette uitvoer te hebben.**

### GDOS ongemak

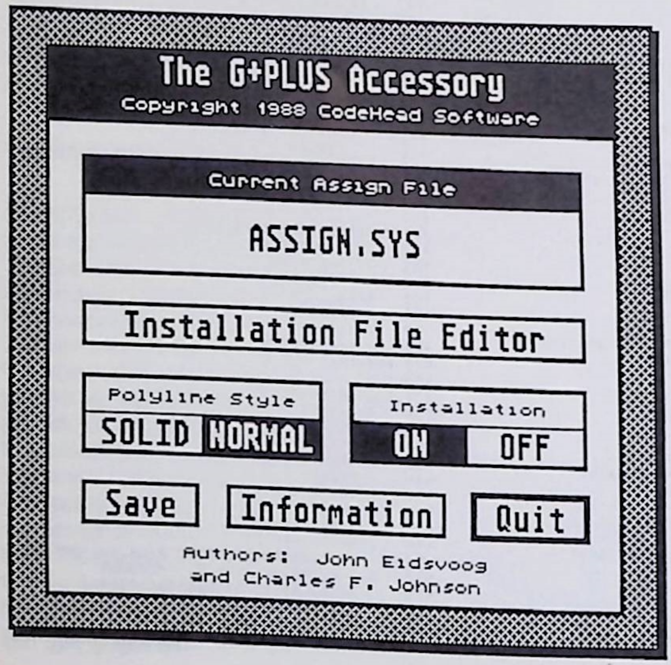
Gebruikers die met meerdere programma's werken die gebruik maken van GDOS, kennen inmiddels de ongemakken van GDOS. Elk programma heeft een eigen ASSIGN.SYS file nodig. Als men gewerkt heeft met TimeWorks en men wil daarna met FastCom werken, dan dient de juiste ASSIGN.SYS ingeladen te worden, omdat FastCom nu eenmaal andere karakters gebruikt dan TimeWorks. Dit kan alleen maar door de computer opnieuw te resetten. Dan spreek ik nog niet eens over het gebruik van verschillende ASSIGN.SYS files bij één en dezelfde applicatie. Zo heb ik voor TimeWorks verschillende ASSIGN.SYS files. Eén om een nette brief te maken en een ander om een pagina van bijvoorbeeld een handleiding samen te stellen. Het combineren van ASSIGN.SYS files heeft al menig ST-gebruiker hoofdpijn bezorgd. Daarnaast zorgt GDOS voor een behoorlijke vertraging van de ST. Verder zijn er een aantal programma's die niet (goed) werken met (een bepaalde versie van) GDOS.

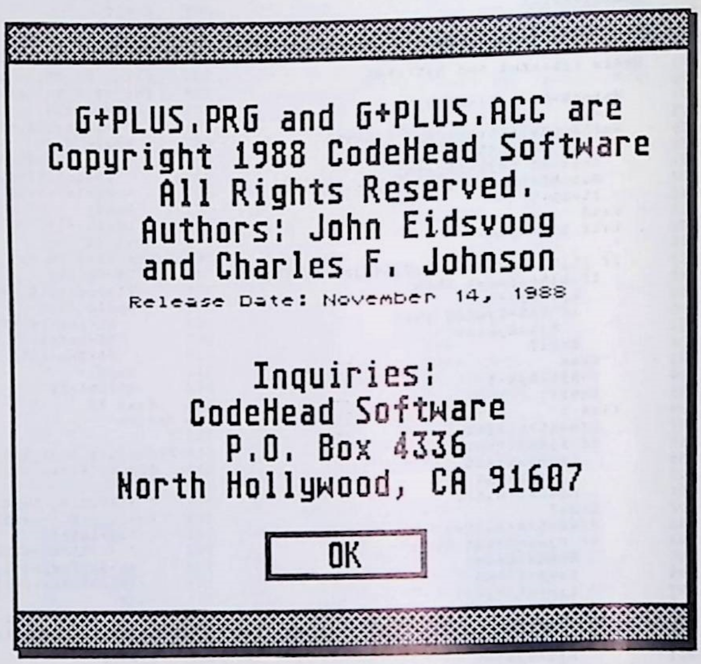
### G+PLUS gemak

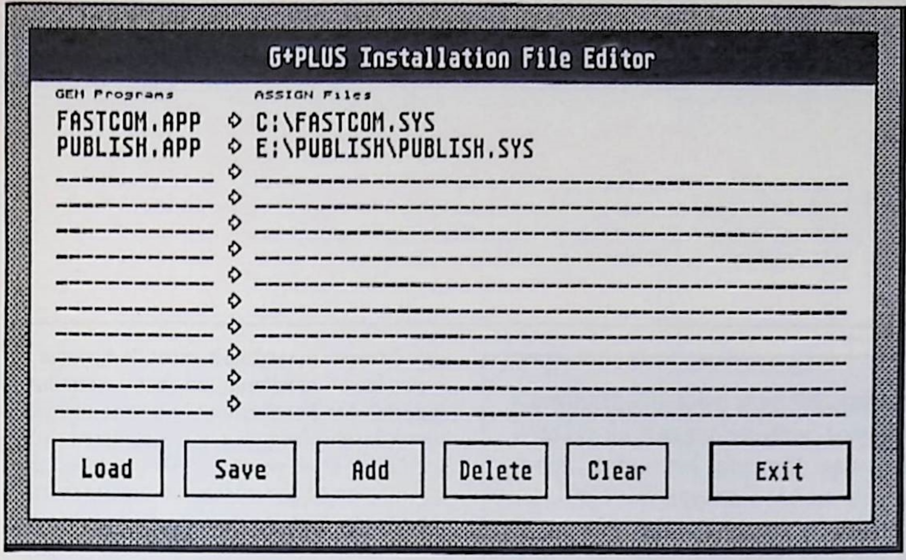
De ongemakken van GDOS die hierboven vermeld staan komen allemaal te vervallen bij het gebruik van G+PLUS. G+PLUS lijkt volkomen onzichtbaar voor programma's die GDOS niet aanspreken. Ik heb niet één keer een programma gehad dat met G+PLUS in het geheugen niet werkte. Verder lijkt elk programma dat gebruik maakt van GDOS met G+PLUS te werken. Van vertraging van de computer is geen sprake. Sterker nog, een aantal grafische functies zijn sterk verbeterd. Het over de desktop verplaatsen van ikonen en het openen van een directory-window gaat merkbaar sneller.

### ASSIGN.SYS Files

Het grootste probleem met GDOS onstaat door de verschillende ASSIGN.SYS files die nodig zijn voor verschillende programma's. De makers van G+PLUS leveren bij het programma een accessory waarmee aan diverse programma's een bijbehorende ASSIGN.SYS file gekoppeld kan worden. Het makkelijkst is om de ASSIGN.SYS file welke bij FastCom hoort in de folder van FastCom te zetten, en deze de naam FASTCOM.SYS te geven. Op deze wijze kan aan elke applicatie een bijbehorend ASSIGN.SYS gekoppeld worden. G+PLUS bewaardt deze tabel in het geheugen. Op het moment dat het programma (b.v. FastCom) aangeklikt wordt, kijkt G+PLUS in de tabel of de naam van het programma voorkomt em laadt in dat geval de bijbehorende ASSIGN.SYS file voordat het programma gestart wordt. De gebruiker hoeft zich geen zorgen meer te maken over al de verschillende ASSIGN.SYS files. Als bij het starten van een programma de ALTERNATE-toets ingedrukt is, dan heeft de gebruiker de mogelijkheid om een willekeurig





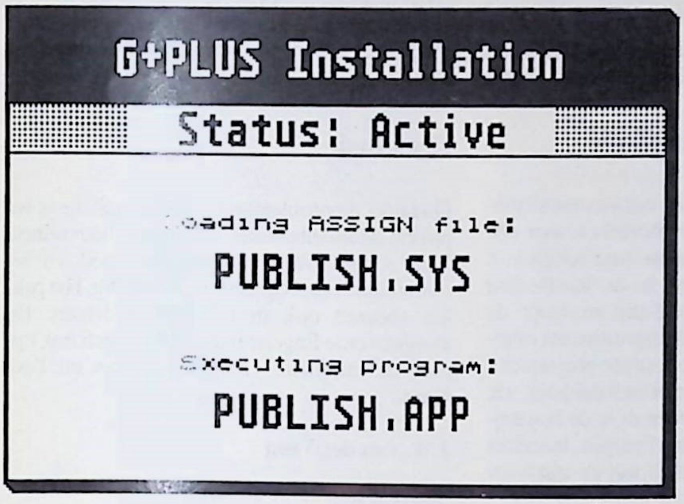

ASSIGN.SYS file te selecteren. Dit is ideaal bij het gebruik van een DTP-programma.

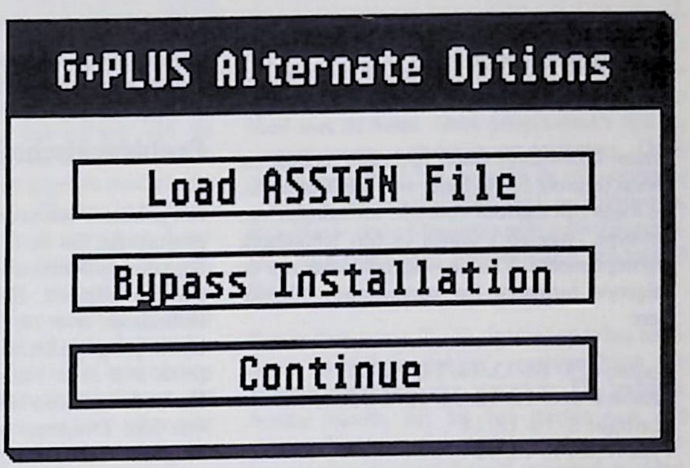
## Geen klachten?

Het gebruik van G+PLUS is zo eenvoudig, dat je na een paar dagen niet eens meer in de gaten hebt dat je het gebruikt. Zeker voor harddiskgebruikers die veel programma's hebben die gebruik maken van GDOS is het een ideale vervanging van GDOS.

Voor gebruikers zonder harddisk is het echter wel even opletten. Als u bijvoorbeeld een diskette heeft met b.v. FastCom en een diskette met b.v. Easy-Draw waarop de ASSIGN.SYS files op beide diskettes daadwerkelijk ASSIGN.SYS genoemd zijn, dan wordt bij het wisselen van programma's niet de andere ASSIGN.SYS file geladen. Dit is op te lossen door elke ASSIGN.SYS file een unieke naam te geven.

**Bert Rozenberg**




*(vervolg van pag. 51)*

## Analyze!

```
445       If Right$(R$,1)=Chr$(32) Then
446         L%=1
447       Else
448         L%=23
449       Endif
450     Endif
451 Return
452 '
453 Procedure Einde
454     ' ==>PRINT CHR$(27);"e"; ' cursor aan
455     ' ==>MOUSEOFF
456     Close
457     End
458 Return
459 '
460 ' ==>DEF FN Exist(F$)
461 ' ==>   LOCAL A,H,R,Ok
462 ' ==>   F$=F$+CHR$(0)
463 ' ==>   A=LPEEK(SEGPTR+28)
464 ' ==>   A=A+ LPEEK( VARPTR(F$))
465 ' ==>   GEMDOS (H,61, HIGH(A), LOW(A),0)
466 ' ==>   IF H>6 THEN
467 ' ==>     Ok=Waar
468 ' ==>     GEMDOS (R,62,H)
469 ' ==>   ELSE
470 ' ==>     Ok=Fout
471 ' ==>   ENDIF
472 ' ==>RETURN Ok
473 '
474 ' ==>DEF PROC Fsel_Screen(R P$,R F$,R F)
475 ' ==>   LOCAL A$,A,Scrn
476 ' ==>   A$=SPACE$(32000)
477 ' ==>   XBIOS (Scrn,3)
478 ' ==>   A= LPEEK( SEGPTR +28)
479 ' ==>   A=A+ LPEEK( VARPTR(A$))
480 ' ==>   MOUSEOFF
481 ' ==>   MEMORY_MOVE Scrn,32000 TO A
482 ' ==>   MOUSEON
483 ' ==>   FILESELECT(P$,F$,F)
484 ' ==>   MOUSEOFF
485 ' ==>   A= LPEEK( SEGPTR +28)
486 ' ==>   A=A+ LPEEK( VARPTR(A$))
487 ' ==>  MEMORY_MOVE A,32000 TO Scrn
488 ' ==>  MOUSEON
489 ' ==>RETURN
490 '
491 ' Voor Omikron alles hierna wissen
492 '
493 Procedure Lees_lregel
494   Local P%,Bl%,T$
495   P%=Instr(Lb$,Chr$(10))        ! zoek LF
496   If P%=0                  ! geen LF gevonden
497     Repeat                         ! herhaal:
498       Bl%=Min(Lfl%-Lfp%,Blk_size%)
499       If Bl%      ! als blokgrootte > 0
500         T$=Lb$
501         Lb$=Input$(Bl%,#1)
502         Lb$=T$+Lb$    ! vul blok aan
503         Add Lfp%,Bl%
504       Else
505         Lregel$=Lb$  ! laatste regel,
506         Lb$=""         ! evt. zonder LF
507         Leof%=True
508       Endif
509       P%=Instr(Lb$,Chr$(10))
510     Until P% Or Leof%     ! tot LF of EOF
511   Endif
512   If P%                    ! Wel LF gevonden
513     Lregel$=Left$(Lb$,P%-2) !zonder CRLF
514     Lb$=Right$(Lb$,Len(Lb$)-P%)
515   Endif
516 Return ! Lregel$
517 '
518 Procedure Lees_rregel
519   Local P%,Bl%,T$
520   P%=Instr(Rb$,Chr$(10))
521   If P%=0
522     Repeat
523       Bl%=Min(Rfl%-Rfp%,Blk_size%)
524       If Bl%
525         T$=Rb$
526         Rb$=Input$(Bl%,#2)
527         Rb$=T$+Rb$
528         Add Rfp%,Bl%
529       Else
530         Rregel$=Rb$
531         Rb$=""
532         Reof%=True
533       Endif
534       P%=Instr(Rb$,Chr$(10))
535     Until P% Or Reof%
536   Endif
537   If P%
538     Rregel$=Left$(Rb$,P%-2)
539     Rb$=Right$(Rb$,Len(Rb$)-P%)
540   Endif
541 Return ! Rregel$
542 '
543 Procedure Zet_buf_globals
544   Blk_size%=512 ! optimale buffer
545   Lfp%=0          ! file pointers
546   Rfp%=0
547   Leof%=False    ! end of file
548   Reof%=False
549   Lfl%=Lof(#1)  ! file lengte
550   Rfl%=Lof(#2)
551   Lb$=""          ! blok buffer
552   Rb$=""
553 Return
554 ' ********** EINDE LISTING 1 **********
```

# Fontpac

**Tot nu toe waren er weinig GEM-fonts beschikbaar voor programma's als Timeworks en Easy Draw. Vooral voor wetenschappelijke teksten met mathematische en Griekse fonts was Signum het enige echte alternatief. Voor deze fontbehoeftigen is er nu het Fontpac van Commedia. Dit pakket is ruim voorzien van fonts van verschillende grootte uit het Metafont pakket (TeX), omgezet naar het GEM-font formaat. Eerst ontvangt de besteller de fonts voor het scherm en 9-naalds printers, daarna kunnen, na het opsturen van het registratieformulier, printer fonts voor de laserprinter (300 DPI) en de 24-naalds printer (180 en 360 DPI) zonder kosten aangevraagd worden.**

## Gecomprimeerde fontbibliotheken

Omdat GEM-fonts raster fonts zijn, nemen ze, vooral bij grote printerfonts, veel schijfruimte in. De fonts zijn daarom voor alle afmetingen van een type (typeface) samen in één bibliotheek gecomprimeerd. Van de volgende fonts zijn de volgende formaten (in puntgrootte) opgenomen:

Roman 6/8/10/12/14/18/24/36
SansSerif 6/8/10/12/14/18/24/36/54
TextItalic 8/10/12/14
TypeWriter 8/10/12
Dunhill 10/12
Inch /100
MathGreek 8/10/12/14/18/24
MathItalic 8/10/12/14/18/24
MathSym 8/10/12/14/18/24

Van de eerste vier fonts zijn ook nog verschillende varianten beschikbaar. Met het programma Fontpac kunnen deze fonts uit hun bibliotheek gedecomprimeerd en bekeken worden. Erg handig hierbij is dat het programma aangeeft hoeveel geheugen het pakket van de gekozen fonts inneemt. Na het decomprimeren is dan bekend of er niet teveel geheugen is bezet en kan het programma zelf een bijbehorende ASSIGN.SYS file voor GDOS genereren. Een heel lastig karwei om met de hand te doen, vooral met zulke hoeveelheden fonts. Ook is het mogelijk eigen fonts op te nemen in nieuwe of bestaande bibliotheken.

## Het resultaat

De fonts zijn heel mooi, ze hebben geen rafels, ook de grootste fonts niet. Als voorbeeld zijn met de Atari laserprinter de mathematische fonts afgedrukt m.b.v. Easy Draw. Deze fonts komen duidelijk uit een wetenschappelijk pakket, bijna alle wiskundige tekens staan er in. De MathItalic is goed voor super- en subscripts te gebruiken.

## Problematische installatie

Het pakket is ook voorzien van een installatieprogramma, dat de fontbibliotheken voor een bepaalde systeemconfiguratie naar schijfjes of partities kopieert. Helaas is de handleiding onduidelijk over de installatie en loopt de beschrijving van het installatieprogramma ongemerkt over in die van het Fontpac programma. Het is de lezer dan ook niet snel duidelijk wie wat doet. Daarnaast worden er in de beschrijving van het menu van Fontpac indexfiles genoemd zonder te vertellen wat en waarvoor indexfiles zijn.

Naast de handleiding is ook het installatieprogramma verwarrend. Het programma dient onder andere als kopieerbeveiliging. Als het voor de eerste maal draait, moet de eigenaar zijn naam laten installeren en het serienummer opgeven. De handleiding verzekert ons ervan hierna de schrijfbeveiliging te installeren en deze nooit meer te verwijderen. Als het programma echter voor de tweede maal draait, moet de beveiliging verwijderd worden, wil het programma niet vastlopen. Het volgende, zeer waarschijnlijke, probleem is de schijfruimte. Voor de gecomprimeerde bibliotheken en het Fontpacprogramma is 1.2 Megabyte nodig. Als deze ruimte niet aanwezig is, loopt het installatieprogramma weer vast, zonder verder enige waarschuwing te geven. Na deze ervaring is het zeker niet aan te raden de fonts automatisch te laten decomprimeren door Fontpac, met de hand kan het proces in de gaten gehouden worden via de geheugenindicator.

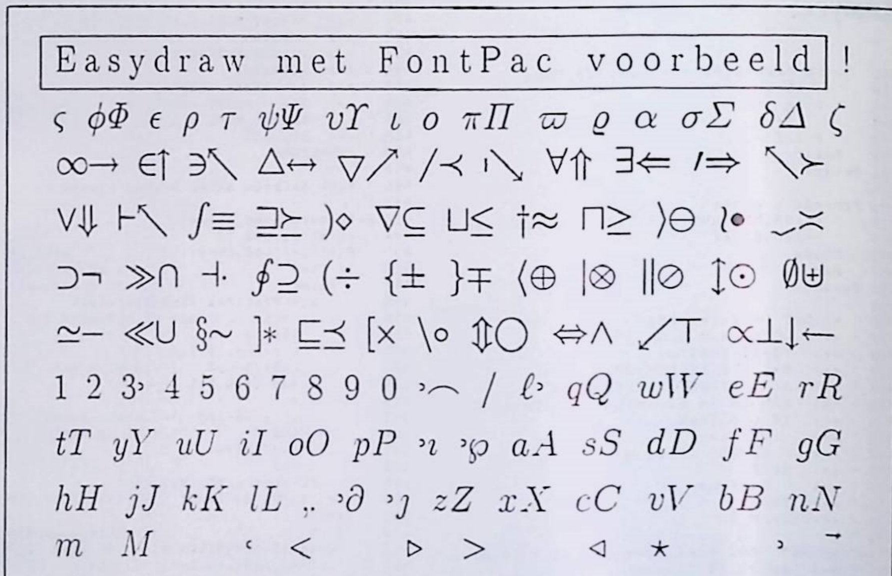
## Conclusie

Ondanks de problemen met de installatie is het pakket de moeite zeker waard. De hoeveelheid fonts is zeer groot, de kwaliteit goed, en het fontbibliotheekprogramma erg handig. Het pakket voorziet ook in GDOS en drivers. De meegeleverde Engelse handleiding geeft ook tips over de installatie voor Timeworks en Easy Draw.

**J.W. van der Veen**

Geleverd door: ComMedia
Leidsekade 98
1017 PP Amsterdam

prijs fl 99,-



# Megamax Modula-2

**Nu de ontwikkelingen op TDI Modula-2 gebied stil lijken te liggen, is het interressant om eens naar een andere Modula-2 te kijken. Eén van de concurrenten van TDI is de versie van Application Systems Heidelberg. Dit pakket bestaat uit een in Duitsland omgeschreven compiler van Wirth. Net als bijna alle nieuwe Modula-2 compilers is dit de 'one pass'-versie geworden. Uit de versie (3.6), maar ook uit de beschrijving van de gevonden fouten, blijkt dat er actief aan het pakket gewerkt wordt. Jammer is dat het een lange tijd geduurd heeft voordat het binnenkwam. Er is in Nederland geen importeur, en liefhebbers zullen dus zelf via Heidelberg Megamax Modula-2 moeten bestellen.**

Het pakket zelf bestaat uit een Duitse handleiding in een ringband met vier schijfjes. Op drie schijfjes zijn de compiler, linker en een eigen editor verdeeld. Ook is een resource construction set in het pakket opgenomen; de al op zich 150 gulden kostende versie van Kuma. Het geheel wordt afgesloten met een schijfje met voorbeeld programma's.

## De handleiding

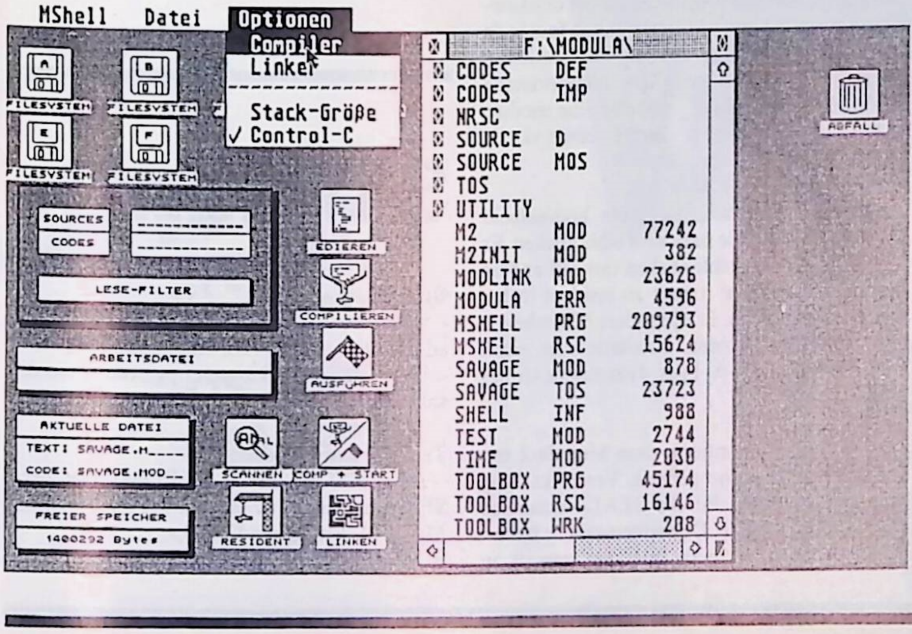
De handleiding is in het Duits, maar gelukkig niet het populaire Duits dat in sommige Duitse tijdschriften gebruikt wordt. De handleiding begint met een inleiding in het gebruik van de shell. Naast het gebruik van de grafische eigenschappen wordt ook de commandostructuur van de opstartfile behandeld. Deze file vervangt de Modula-2 options accessory van TDI. Andere instellingen kunnen in de shell gemaakt worden. De beschrijving van de compiler en linker is kort; alleen het gebruik komt aan de orde. Ook kort is de beschrijving van de debugger, een onderdeel van het pakket waar toch meer informatie over gegeven mocht worden. Uitgebreid gaat de handleiding in op de in de programma's op te nemen assembler. Twee van de eigenschappen die behandeld worden, zijn het integreren van in Modula-2 gedeclareerde variabelen in een assembler-gedeelte en het aanroepen van andere Modula-2 procedures. Daarna volgt een uitgebreide behandeling van de geleverde bibliotheken. Deze komen alle uitgebreid aan de beurt, wat een hele verbetering is ten opzichte van de gangbare korte definitiemodulen. Behandelde bibliotheken zijn bijvoorbeeld de Line-A, GEM en de systeembibliotheken. De handleiding wordt afgesloten met de definitiemodulen en een korte beschrijving van de Kuma resource construction set.

## De shell

Alle handelingen, zoals schrijven, compileren en linken, gaan via de shell. Deze is grafisch georiënteerd en heeft voor de verschillende handelingen verschillende iconen. Ook de typische desktophandelingen zijn mogelijk met de drive-icons. In het voorbeeldplaatje is deze alternatieve desktop te zien met een geopend window voor drive F. Als de shell niet geheel aan de wensen voldoet, is deze aan te passen, want de programmatekst is meegeleverd.

Eén van de prettige eigenschappen van de shell is de mogelijkheid om veel gebruikte programma's resident in het geheugen op te slaan. Dit is automatisch te doen via de opstartfile of via de shell met de hand. Deze programma's zijn nu vrijwel zonder vertraging op te roepen. Door deze constructie is het mogelijk de edit - compile - edit-cyclus comfortabel snel te doorlopen. Behalve de door Megamax geleverde programma's zijn ook eigen programma's resident te maken.

Een andere versnellende eigenschap is het zogeheten 'load-time- linking', waarvan bijna alle nieuwe pakketten zijn voorzien. De techniek bestaat hierin, dat bij het starten van een gecompileerde module de geïmporteerde modulen geladen en tegelijkertijd gelinked worden. Megamax Modula-2 voorziet bovendien nog in de mogelijkheid om in de opstartfile op te geven welke modulen standaard al ingeladen moeten worden. Het laden van de bibliotheek modulen vervalt en het programma start vrijwel direct.




## De editor

Megamax levert een eigen editor waarvan ook de C programmatekst verkregen kan worden. Dit kost dan wel weer een slordige 150 DM.

De editor is zeer uitgebreid, waardoor de menu's overvol zijn. Er is zelfs een menu opgenomen om via de modem source's binnen te halen. Megamax heeft daarvoor een eigen BBS, Maus, die echter wel in Duitsland gelocaliseerd is. Niet alleen met de overvolle menu's maar ook met andere eigenschappen onderscheidt deze editor zich van bijvoorbeeld Tempus. Een van de direct opvallende verschillen is, dat de Megamax editor geen gebruik maakt van windows. Om toch aan meerdere teksten te kunnen werken, is het mogelijk tussen verschillende schermen te wisselen. Een eigenaardigheid hierbij is dat voor het tweede venster het maximaal aantal regels dient opgegeven te worden. Mocht de ingeladen

tekst langer zijn, dan wordt deze afgebroken. Een bijkomend kritiekpunt is dat een derde en volgend venster niet meer regels kan bevatten dan het tweede.

De snelheid van de editor is uitstekend. Dit komt waarschijnlijk vooral doordat de regels op 80 karakters vast gezet zijn. Een gevolg daarvan is dat de cursor ook na een regeleinde vrij bewegen kan.

Alle commando's zijn ook via het toetsenbord op te roepen. Ook de compiler kan van het toetsenbord vanuit de editor opgeroepen worden. Direct na een programmeerfout geeft de compiler een alertbox waarmee weer naar de editor gesprongen kan worden. Deze cyclus is razendsnel uit te voeren. Een minpunt hierbij is dat het waarschijnlijk niet mogelijk is het programma in één keer op alle programmeerfouten te bekijken.

## Compiler en linker.

Deze programma's zijn als gecompileerde modulen beschikbaar en alleen uit de shell op te roepen. De compiler is een 'one pass' versie en voor later gedeclareerde maar eerder gebruikte procedures kan de FORWARD-declaratie gebruikt worden. Hiermee wordt de compiler verteld dat de procedure verder in de programmatekst te vinden is. De linker is een optimaliserende linker. Overbodige procedures worden niet in de uitvoerbare code opgenomen. Op deze wijze worden programma's die enkele procedures uit grote bibliotheken halen niet onnodig vergroot. Als hoogste optimalisatie is het mogelijk de procedurenamen niet in de code op te nemen. Het nadeel hiervan is dat tijdens het vastlopen van het programma niet aangegeven wordt welke procedure de boosdoener was. Ook is het mogelijk de uitvoer via de module InOut om te leiden naar een window. Er moet dan de module GEMIO in plaats van TOSIO meegelinked worden.

## Debugger

Het is niet best gesteld met de debuggers voor Modula-2. Als een programma vastloopt, is meestal wel te zien waar het fout ging. Echter, zodra er door het programma gelopen moet worden om het verloop van een variabele na te gaan, moeten we zelf wat routines in elkaar draaien. Megamax biedt nu een echte debugger waarmee door de programmatekst gestapt kan worden. Na elke stap geeft de debugger de tussenresultaten van berekeningen en de eventuele resultaten van toekenningen. Verdere mogelijkheden om variabelen te bekijken zijn er niet, maar het voldoet. Ook laat de debugger de parameters bij een procedure-aanroep zien en kunnen met een commando de CPU-registers zichtbaar gemaakt worden. In twee plaatjes staan voorbeelden van een savage testprogramma. Uitvoer in een programma wordt in het StdIO window uitgevoerd.

## De assembler

Met het woord ASSEMBLER kan er in assembler geprogrammeerd worden. Een door Megamax gegeven voorbeeld om een bitreversal te doen op k is:

```
FOR k:=0 TO kMax DO
  ASSEMBLER                ; *** kRev := BitReversal(k)
       MOVE  k,D0          ; get loop index
       MOVEQ #15,D1        ; 16 bits are to be reversed
   lp  LSR   #1,D0         ; shift a bit in the extend flag
       ROXL  #1,D2         ; .. and to the destination reg.
       DBF   D1,lp         ; 16 times
       MOVE  D2,kRev       ; move the result to var. kRev
  END;
...
END (* FOR k *)
```

*Listing 1: Assembler gebruik in Megamax Modula-2.*

Zoals te zien is, is het gebruik van in het Modula-2 gedeelte gedeclareerde variabelen heel eenvoudig.

## Ervaringen met Megamax Modula-2

Ervaringen zijn erg persoonlijk dus moet iedereen zelf maar beslissen hoe belangrijk ze zijn. Eén van de dingen die direct opvallen is de bijzonder snelle cyclustijd. Als er wat veranderd moet worden, draait het programma in een oogwenk. Ook de editor is, vergeleken met de TDI-versie, een enorme verbetering. Degenen die echter gewend zijn aan de luxe van Tempus, moeten zeker wennen aan het andere 'gevoel' en andere commado's.

Een echt vervelende eigenschap van het pakket is de naamgeving van de GEM-bibliotheken. De schrijver van deze bibliotheken heeft kosten noch moeite gespaard om alle procedures anders (nl. met omkering van woorddelen) te noemen dan in de TDI-versie. Bovendien zijn de variabelen aangepast om ze meer in het Modula-2 jasje te laten passen. Dat is niet overal goed gelukt; vooral met de object-bibliotheek is het aanpassen weer losgelaten. Daarnaast zijn typedeclaraties van één bladzijde (variant record) ook niet ideaal.

Een andere vreemde eigenschap is het onnauwkeurig aangeven van de plaats van een fout in de programmatekst. Een dubbele declaratiefout tussen een in de import lijst geïmporteerde procedure en één die al in de definitie module gedeclareerd is, wordt pas aan het einde van de import lijst aangegeven.

Een groot pluspunt is de grote hoeveelheid bibliotheken naast de standaard bibliotheken. Er zijn onder andere bibliotheken om zelf andere programma's aan te roepen en resident in het geheugen te houden. Er zijn andere bibliotheken voor file management, installatie van eigen trap-handlers, Line-A, Lists, dynamische strings en veel meer.

Ondanks de minpuntjes is deze Modula-2 een hele verbetering in het gebruik. Verder kan nog opgemerkt worden dat alle REAL- variabelen standaard 64 bits zijn. De mathematische bibliotheek is erg snel, maar niet erg precies in vergelijking met andere Modula-2 pakketten. Deze test zal de volgende keer in het blad verschijnen als test van een aantal pakketten.

**Jan Willem van der Veen.**

*Megamax Modula-2*
*prijs DM 398,-*
*Application Systems Heidelberg*
*Englerstraae 3*
*D-6900 Heidelberg*
*tel. (0 6221) 30 00 02*

# ST markt

**Gratis:**
- voor abonnee's is het plaatsen van kleine advertenties in deze rubriek geheel gratis.
- kleine annonces kunnen opgestuurd worden aan het secretariaat.

**Te koop aangeboden:**
- Atari 520 ST (TOS in ROM) met drive SF354 en muis. (i.v.m. met aanschaf van een MegaST). Vraagprijs Fl 695,-.
R. Hilbrandie, tel: 050-412276.

# SPC-MODULA

**Sind enige maanden is er in Duitsland een Modula ontwikkelingspakket te krijgen van de firma Advanced Applications Viczena uit Karlsruhe. Nu is het moeilijk om na een week gebruik al diepgaand op een dergelijk product in te gaan, maar een globale beschrijving van de geleverde software en een voorlopige indruk zijn wel te geven.**

Het Modula-pakket zoals dat hier verder besproken wordt, is de laatst uitgebrachte versie 1.4 van februari 1989 en kost ongeveer 400 DM. Voor die prijs krijgt men een handboek in ringband van ongeveer 400 bladzijden, waarvan de helft wordt ingenomen door de beschijving van de definition modules van de bijgeleverde bibliotheken, en drie dubbelzijdige disks. Op de disks staan een compiler, een linker, een editor, een shell, een debugger, tal van utitities (tools), een aantal bibliotheken met modules en een Modula-cursus afkomstig van de TU te Muenchen.

Om goed met het pakket te kunnen werken is minimaal 1 Mbyte geheugen nodig, een monochroom monitor, en twee dubbelzijdige diskdrives. Er is pas goed en vlot te werken als er een harddisk is met minimaal 2 Megabyte vrije opslagruimte.

In Duitsland is er een kaart te krijgen, de PAK-uitbreiding, die is uitgerust met een 16mHz 68020 en een floatingpoint chip, de 68881. Men kan ook een versie van het ontwikkelingspakket krijgen (voor 50 DM extra) die in staat is om code voor een dergelijke opgevoerde ST te maken (met die kaart lopen programma"s die veel gebruik maken van GEM een faktor 1,5 sneller, en voor andere toepassingen gaat alles zeker vijf maal zo snel.)

## SSWiS

Het bovenstaande is een afkorting van Small Systems Windowing Standard, wat zoveel veel wil zeggen als dat er een groep van functies bestaat, toegesneden op de mogelijkheden van kleine computers, om met vensters te kunnen werken. In hoeverre deze standaard elders in de wereld op andere machines dan de ST is geïmplementeerd weet ik niet. Mogelijk is dit de zoveelste window-standaard.

De mogelijkheden van GEM worden gebruikt om dit interface te ondersteunen. Maar GEM is beperkter dan SSWiS, zodat er wat zaken moeten worden omzeild. Zo moet er een oplossing zijn voor het feit dat SSWiS enige tientallen vensters gelijktijdig in beeld kan hebben, maar dat GEM het met een handjevol wel heeft bekeken.

Gekozen is voor het afbeelden van vensters die 'teveel' zijn in de vorm van tekst ikonen. Aanklikken van zo'n ikoon zet dan het venster dat erbij hoort op het scherm, maar zijn er al teveel vensters open (voor GEM), dan zal het meest onderliggende scherm verdwijnen, en omgezet worden tot een tekst-ikoon op de Modula- desktop.

## De shell

In dit systeem werkt men als programmeur binnen een shell (schaal) die zich presenteert als een desktop compleet met menu's, vensters en tekst-ikonen. De turn-around tijd is kort, wat komt doordat de compiler snel is, maar bovendien, doordat er gebruik gemaakt wordt van een zogenaamde Loader die zorgt voor de uitvoering van programma's. De activiteiten van de loader worden wel eens run-time linking genoemd. Wat namelijk gebeurt is dat programma's in de vorm van objekt modules worden uitgevoerd (dus in de vorm zoals ze uit de compiler komen), en dat, als tijdens de uitvoering een ander objektmodule nodig is, dit door de loader opgezocht en ingeladen wordt en gebruikt. Aangezien de compiler zelf niets weet (tijdens het vertalen van een programma) van de plaats waar de gemaakte code in het geheugen zal komen te staan, en ook niet weet waar aanroepen naar functies van andere (afzonderlijk gecompileerde) modules in het geheugen (zullen) staan, is het de opgave van de loader om dit op te lossen tijdens het draaien van het programma.

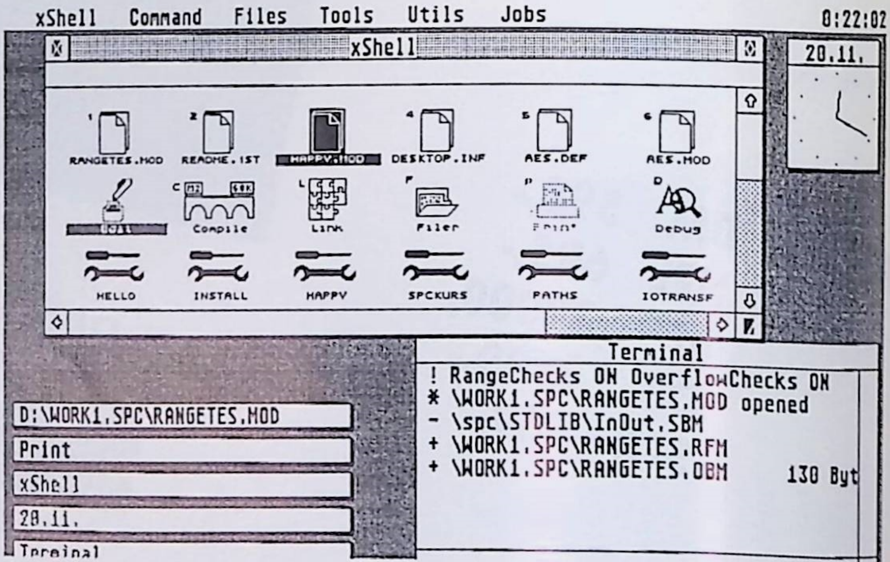
De grafische interface van de shell werkt met het SSWiS interface, zodat er een soort multitasking mogelijk is. Het is daardoor erg makkelijk om zelf gemaakte programma's als utilities in de shell in te binden.

## De editor

De editor is speciaal gemaakt om met Modulateksten te werken. Het is mogelijk om meerdere vensters open te hebben op diverse files. Dit is handig, want het komt nogal eens voor dat men b.v. de tekst van een definitie-module moet inzien bij het construeren van een programma.

Als er tijdens compilatie een fout is gevonden, keert men automatisch in de editor terug, met de cursor op de plaats van de eerst optredende fout. Doorspringen naar eventuele verdere fouten is mogelijk met de editfunctie 'error-next'.

De werking en bediening van de cursor zijn niet afwijkend van andere editors; een regel wordt b.v. gewist met de shift-delete toetsencombinatie. Ook de Undo-toets werkt (tot twintig handelingen terug!), wat maar bij weinig editors het geval is. Het is mogelijk om in het venster het regelnummer te zien waarop de cursor staat. Nagenoeg alle functies van de editor zijn zowel



via de muis als via het toetsenbord te bereiken. Ook operaties op blokken tekst zijn mogelijk. Een extra is hierbij dat het mogelijk is om tekstblokken niet alleen in vertikale richting te verplaatsen, maar ook in horizontale richting. Dat is handig in verband met het inspringen van Modula-teksten.
Het numerieke toetsenbord (het rechter deel met de getallen) kan gebruikt worden voor het opbergen van tekst strings, die de editor dan kan invoegen in de tekst die onder bewerking is.
Een extra mogelijkheid van de editor is een taalondersteuning, die bijvoorbeeld ervoor zorgt dat het intypen van IMPL al voldoende is om het woord IMPLEMENTATION in de tekst te laten zetten. Het intypen van PROC geeft dan het woord PROCEDURE, lastig als je echt PROC bedoelde, maar gelukkig kan dit feature worden afgezet.

## De Compiler

De compiler is van het one-pass type, en is afgeleid van de basisversie die ontwikkeld is aan de ETH van Zuerich. Deze compiler levert 68000-code en gaat door Modula-teksten heen met een snelheid van enige duizenden regels per minuut. Deze snelheid loopt nogal terug als er veel imports worden gedaan, maar is nog altijd groot genoeg voor de meeste mensen. Ter vergelijking: op een VAX mini-computer haalt de VAX-VMS Modula-compiler ongeveer 1100 regels per minuut, en als je dan met 5 mensen aan een VAX zitten, kosten zoveel regels je dus 5 minuten wachten.

## De Linker

Hoewel het handig en snel is om Modula-programma's te draaien onder de shell, zal op een gegeven moment een programma ook stand-alone moeten kunnen werken; er moet uiteindelijk een .PRG o.i.d. gemaakt kunnen worden. De linker is het stuk gereedschap om dit te doen. Deze zoekt uit welke modules gebruikt gaan worden, test of er geen conflicten zijn (de version-conflicts) en plakt de machine instructies aaneen tot een werkend programma. Van nature is de werking van een linker zeer langzaam. Veel files moeten in en uitgeladen worden, en veel adresberekeningen moeten verricht worden. De linker kan, in zijn huidige versie (1.4), nog niet optimaliseren. Hierdoor worden programma's nogal groot, en zitten vol met machinecodes die nooit gebruikt worden door het programma. Het is lastig dat het gebruik van een enkele variabele of functie uit een module tot gevolg heeft dat alle data en code van het module in je uiteindelijke programma terecht komen. Zo wordt de minimale grootte van een programma zonder GEM ongeveer 10 Kbyte, en met weinig GEM al gauw 100 Kbyte.
Momenteel wordt er hard aan deze linker gewerkt en zal er bij de nog te komen versie 1.5 geoptimaliseerd kunnen worden.
De linker is van het two-pass type. In de eerste fase worden alle modules die nodig zijn opgezocht, en wordt er gekeken of de module sleutels kloppen (in verband met mogelijke 'version conflicts'). In de tweede run worden de data uit de files gehaald en behandeld (dit plak en patch werk is het eigenlijke linken), vervolgens wordt er nog een speciaal run-time systeem toegevoegd, en het resultaat wordt in een file van het type .PRG weggeschreven. Men heeft dan een kant en klaar programma, dat volledig op zichzelf kan draaien zonder ondersteuning van een Modula-omgeving (shell).
Er blijkt geen mogelijkheid te zijn om accessoires te maken, maar in een volgende versie is dit manco waarschijnlijk opgeheven.

## De 32 Kb limiet

De compiler (en ook de linker) zijn niet in staat om meer dan 32 Kbyte aan te kunnen. Zo moet de code van de gecompileerde definitie-, implementatie- en programma-modulen altijd kleiner zijn dan 32 Kbyte. Ook array's, record's en andere datastructuren hebben eenzelfde limiet op de grootte. De makers van het pakket hebben beloofd dat in de volgende versie deze beperking opgeheven zal zijn.

## De 16-32 bit kwestie

Verwant aan het 32 Kbyte-limietprobleem is de kwestie 16 en 32 bits. Onze CPU is van een gemengd 16/32-bits type. Adressen zijn 32 bits groot (waarvan er maar 24 door de hardware worden gebruikt), dus POINTERS zijn 32 bit. De snelst toegangkelijke geheugeneenheid is het 16 bits woord. Dus logischerwijze worden INTEGER en CARDINAL geïmplementeerd als 16 bit, en zijn dan niet uitwisselbaar met POINTERS. Vandaar dat er ook altijd LONGCARD en LONGINT moeten zijn, die wel compatibel zijn met adressen.
Verder werkt de CPU zeer efficiënt bij sprongen in de code als er gerekend kan worden met offsets van 16 bits ten opzichte van een basisadres. Maar moet er verder worden gesprongen, dan zijn er extra CPU-instructies nodig om de adresberekening te doen, en wordt het gauw viermaal zo langzaam. Om de 32 Kbyte-limiet van compiler en linker kwijt te raken, moet er dus heel wat intelligentie worden bijgebouwd.

## Low level

In deze versie kan er met behulp van de INLINE-functie gebruik gemaakt worden van machinecode in Modula-programmateksten. In de toekomst zal er een interface komen, waarbij men in C (als een soort algemeen assemblertaaltje) kan programmeren.
Momenteel is er een module in de bibliotheek dat in staat is om een stuk machinecode om te zetten in INLINE-statements, zodat men geen fouten maakt omdat men zaken zit in te typen vanaf papier.

## Snelheid

Een van de punten waarop compilers worden beoordeeld, is de snelheid waarmee de resulterende code werkt. De resultaten van wat benchmarks laten weinig verschil zien tussen SPC-Modula en Megamax Modula. Wel is opvallend dat integer-berekeningen in SPC-Modula zeker tien keer sneller zouden gaan, een merkwaardig groot verschil dat ik niet kan plaatsen. Het verschil voor bewerkingen op reals is een factor twee in het voordeel van SPC. Evenwel zijn procedure aanroepen in SPC wat langzamer dan in Megamax Modula en hetzelfde geldt voor het ophalen van elementen uit een tweedimensionaal array. Een vergelijk van I/O snelheden is altijd een beetje lastig, maar een verschil met een faktor 15 voor het schrijven van bytes in een file op harddisk ten nadele van Megamax is wel erg moeilijk verklaarbaar.

## Debugger

De bijgeleverde debugger is van het Post-Mortem type. Dus alleen als een programma vast loopt op een (modula)run time error, dan kan er gekeken worden waar het programma vast liep, en wat er dan aan de hand was. Hiertoe krijgt men in een tekstvenster aangewezen in welke source en in welke regel het gebeurde, een ander venster geeft de mogelijkheid om de inhoud van variabelen te inspecteren, en weer een ander venster geeft aan in welke volgorde procedures waren aangeroepen.
Door zelf in programma's een HALT te zetten kan men wel een soort break-point nadoen, maar het blijft behelpen. Een echte run-time debugger is er nog niet (voorlopig).
Bedenk wel dat het merendeel van de fouten in de praktijk gemaakt wordt met het gebruik van het interface naar GEM. Gaat GEM de mist in, dan is er weinig emplooi voor de post-mortem debugger.

## Verdere utilities

Om het werken voor de programmeur aangenamer te maken zijn er diverse handige gereedschappen bijgeleverd. De shell is al genoemd. Hieronder zal ik enige utilities bespreken die vanuit de shell zijn te gebruiken.

## Make

Deze utility is bekend uit de UNIX-wereld, en is vooral handig als men werkt aan grote applicaties. Wat de 'make' doet is vaststellen welke modules voor hercompilatie in aanmerking komen wegens veranderingen in definitie modules. Iedere Modula programmeur kent wel de melding van de linker 'version conflict' als men vergeten is dat een module een versie van een (inmiddels) veranderde definitie importeert. Deze Make-utility werkt aan de hand van het datum/tijd stempel van de files. Men dient dus altijd ervoor te zorgen dat datum en tijd goed staan als men met dit gereedschap op de ST werkt.

## Prelink

Als men een aantal modules heeft gemaakt waarvan bijvoorbeeld zeker is dat, als je er een van gebruikt, je ze alle zal gebruiken, dan kan het handig zijn tijdens verder ontwikkelwerk om deze modules in gecompileerde vorm bij elkaar te zetten in een file. Dat spaart een hoop tijd bij het laden door de Loader. Zo bestaat de meege-

leverde editor uit een aantal modules, die in een enkele file zijn opgeborgen.

## Print

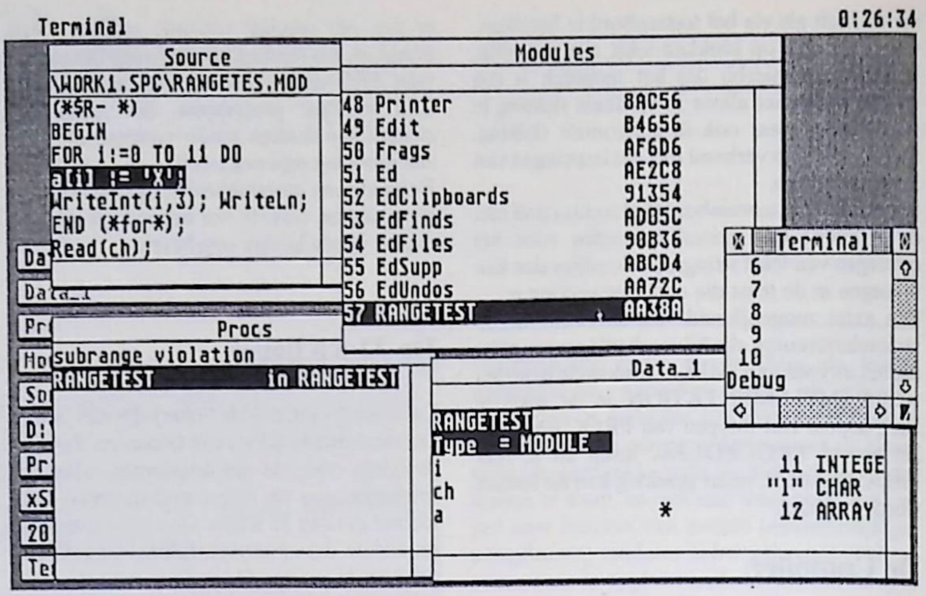
Vanuit de shell kunnen modules uitgeprint worden. De keywords van de Modula-taal kunnen dan in de listing vet afgedrukt worden, tenminste als de printer dat kan. Deze utility heeft een handige oplossing voor het aanpassen aan de printer. Men kan namelijk de printer-installatiefile (de .CFG) van First Word Plus gebruiken. Er zijn heel wat programma's gemaakt die listings en teksten uitdrukken, maar dit is de eerste keer dat ik deze inventieve oplossing tegenkom voor het configuratieprobleem.
Verder kan men via de menu's van deze utility opgeven hoe breed listings mogen worden, of de regels genummerd moeten worden en dergelijke.

## Jobs

Onder Unix kent men een Job Control Language die gebruikt wordt om systeemopdrachten te automatiseren. Meestal gaat het om simpele volgorden als 'compileer file ...', vervolgens 'file... deleten' en 'file... copiëren naar ...' enz. Het taaltje waarin dit onder Unix wordt beschreven is een vorm van kleuter-C.
Op de vroegere Sinclair QL bestond er een zeer uitgebreide Job Control Language die 'Basic' werd genoemd en bovendien toen al in staat was om alles te doen wat men tegenwoordig in Basic kan doen.
In dit pakket wordt (hoe anders?) de Modula-taal tevens als JCL gebruikt. Er is in de bibliotheken een module met de naam JCL, die de faciliteiten bevat voor het copiëren van files, starten van programma's en dergelijke. Een belangrijk gevolg van de hier beschreven opzet is dat jobs zich niet van buiten onderscheiden van utitities, en kan een job dus ook als utility geïnstalleerd worden. Verder staan alle operatoren uit JCL ter beschikking voor 'normale programma's. Konkreet houdt dit in, dat een file-copy vanuit een eigen programma zeer simpel tot stand komt doordat een tekststring wordt doorgegeven aan een JCL-functie.

## Multi-tasking

De taal Modula zelf bevat de benodigdheden voor multi-tasking, doordat er 'Processes' bestaan waartussen geschakeld kan worden. Ook in SPC-Modula is deze mogelijkheid geïmplementeerd. In het verleden zijn er diverse programma's geschreven in TDI-Modula waarbij diverse processen (quasi-gelijktijdig) dingen doen op het scherm. Op onze ST zitten we met het probleem dat zowel het operatingsysteem en ons GEM niet gemaakt zijn voor multi- tasking. Het enige wat ons op dit gebied geboden wordt, zijn de accessoires. De reden dat ik in deze bespreking speciaal de multi-tasking vermeld, is dat het pakket geleverd wordt met een grafisch interface met de naam SSWiS, dat de gebruiker een simpele manier verschaft om de in- en uitvoer van taken te hangen aan vensters op het beeldscherm. De gebruiker schakelt dan tussen de




taken door afwisselend vensters actief te maken met behulp van een muisklik ( Met het window-interface dat in het Brabantse wordt ontwikkeld door Juul van der Spek - zie de uitgave van ST nummer 17 - kan dit ook; de opzet van SPC is op veel punten vergelijkbaar.)

## Cursus

Er wordt een Modula-cursus op disk meegeleverd met dit pakket. Deze cursus is natuurlijk in het Duits en is gemaakt aan de TU van München. Uit dezelfde bron is het PD-Modula afkomstig (zie elders in dit blad en in de lijst van nieuwe PD-software.), en het is geen toeval dat beide Modula ontwikkelpakketten op elkaar lijken als twee zussen van dezelfde moeder.

## Resources

Bij dit pakket is geen Resource Maker aanwezig in de vorm van een RSC (Resource Construction Set). Bij de toolkit van TDI-Modula zat een RSC van Megamax. Maar zo langzamerhand moeten overal wel copieën rondwaren van de resource maker van DRI, die opgenomen was in het oorspronkelijke sofware-ontwikkelingspakket van Atari zelf.

## Support

De ervaringen die mensen hebben met ondersteuning bij problemen is zonder meer goed. Men is bij de firma Advanced Applications bereid om naar de klanten te luisteren. In het handboek vindt men zelfs een formulier om gevonden bugs te melden! Standaard kan met op zijn Modula-desktop (in de shell) een communicatieprogramma vinden, dat ervoor geschikt is om de mailbox van de firma Advanced Applications te bereiken via de telefoon.
De indruk bestaat dat er hard wordt gewerkt aan verbeteringen. Zo zijn de volgende dingen beloofd voor de toekomst:
opheffing van de 32 Kbyte-limiet op data en code;
aanpassing aan de komende (ISO-) normering van de taal;
beter gebruik van de CPU-registers;
de linker zal kunnen optimaliseren;
er zullen accessoires gemaakt kunnen worden en misschien zelfs wel een runtime-debugger.

## Muis en toetsen

Een van de goede dingen van dit pakket is dat alles wat er met de muis kan worden aangeklikt op het scherm (menu-items, iconen e.d.) ook konsekwent door gedefinieerde toetsen op het toetsenbord kan worden bereikt.
Zeker voor mensen die dit pakket intensief gebruiken, is het na een tijdje mogelijk om te onthouden waar allerlei commando's zich op het toetsenbord bevinden, zodat men zijn handen op het toetsenbord kan houden.

## Contactblad

Op onregelmatige tijdstippen geeft de firma een gratis blad uit voor alle geregistreerde gebruikers. Hierin komen bijdragen van gebruikers, voorstellen voor verbeteringen en bugs ter sprake. Verder is er ook nog een mail-box die gebruikt wordt ter communicatie met gebruikers (support) en voor uitwisseling van software.

## Grote beeldschermen

Voor het gemak van de software-ontwikkelaar is een goed beeldscherm geen luxe. Degenen die vele duizenden guldens willen uitgeven aan een MATRIX 110 scherm (19 inch) bijvoorbeeld, hoeven zich geen zorgen te maken over de bruikbaarheid van dit pakket; dat werkt dan ook prima.

## Zwakke punten

Het grafische interface is tamelijk langzaam. Dit wordt speciaal merkbaar als er snel wordt ingetypt in de editor.
Verder is er iets met de muisgevoeligheid in dit SSWiS systeem. Ikzelf heb geregeld moeite om mijn muisklikken effektief te krijgen. Ergens zit er 'dode tijd' in de bediening die de reflexen van de gebruiker een vaagheid geven. De indruk

*(vervolg op pag. 62)*

# Een Public Domain Modula-compiler

**Terwijl er reeds geruime tijd in de PD diverse Assemblers aanwezig zijn, er sinds enige tijd ook een C-compiler is, terwijl er zelfs Modulair Pascal is en het Omikron Basic voor weinig geld te krijgen is, is er nu uiteindelijk ook een compleet Modula-2 pakket verschenen in de PD.**

Het uit Duitsland afkomstige pakket is gemaakt aan de Technische Universitaet van Muenchen, en is daar in gebruik voor het studentenpraktikum.
Het geheel wordt op twee dubbelzijdige disks geleverd, waarvan de eerste disk het Modula-gedeelte bevat en de tweede disk diverse onontbeerlijke zaken (o.a. editors) en handleidingen.

## Oorsprong

In een ver verleden heeft Wirth een compiler geschreven (multi-pass) voor de oorspronkelijke Modula-2 machine, de Lilith. Deze is later omgewerkt tot een one-pass compiler, die veel sneller werkte. Ook de laatste taalveranderingen zijn in die compiler verwerkt. Deze in Modula geschreven compiler is natuurlijk te porteren op de ST, want daar is (b.v.) TDI-Modula beschikbaar. In dit geval is dit dan ook gedaan.

## Loader

Gaat men dit PD-Modula gebruiken, dan roept men als eerste de loader aan. Deze creëert als het ware een schil (shell) waarbinnen men als programmeur Modula-programma's edit, compileert, debugt en runt. De hoofdtaak van de loader is dus het binnenhalen en interpreteren van programma's in objektvorm, dat wil zeggen in de vorm zoals de compiler ze vertaald heeft afgeleverd. De uitdrukking 'run-time Linking' wordt wel eens gebruikt.
Als gebruiker is men erg beperkt door het feit dat Modula programma's met dit systeem enkel binnen de loader gedraaid kunnen worden. Maar gelukkig is er een echte Linker!

## Linker

De linker maakt het mogelijk om stand-alone programma's voor de ST te maken. Dus echte .PRG's en .TTP's zijn mogelijk. Evenwel is er (nog) geen voorziening om accessoires aan te maken (.ACC's). De linker zoekt de benodigde modules op en bindt ze aan elkaar tot een werkend stand-alone programma. Er wordt wel een run-time systeem meegebonden, waarbij de minimum grootte van een programma 7Kb is. De linker is helaas niet in staat om de code te optimaliseren; het importeren van een module heeft tot gevolg dat de complete code van het module wordt meegelinkt, niet de paar gebruikte functies alleen. Zodoende krijgt men bij gebruik van een paar GEM-functies al direkt een fors programma.

## Debugger

Een nagenoeg onontbeerlijk stuk gereedschap is een behoorlijke debugger, die de gebruiker in staat stelt om op Modula-sourcecode niveau zijn fouten op te sporen. Op de disk is een debugger te vinden die onder de loader automatisch in werking treedt als het geteste programma vastloopt. Het is dus een post-mortem debugger en niet een run-time debugger. Dat geeft natuurlijk zo zijn beperkingen, maar er zijn in de wereld bijzonder weinig run-time debuggers die op sourcecode-niveau werken. (Voor Modula op de ST zou enkel Haenisch Modula dit hebben; alle andere softwarehuizen 'zijn eraan bezig'.)

## Editor

Voor het gemak van de programmeur is een goede, maar vooral snelle en compacte editor nodig. In dit pakket zitten twee verschillende editors; men kan dus kiezen.

## De bibliotheken

De bruikbaarheid van een Modula-implementatie op de ST hangt af van de beschikbaarheid van bibliotheken. De standaard bibliotheek van Wirth is aanwezig (met Terminal, InOut, WindowBase enz.) en alle systeeminterfaces worden ook meegeleverd (naar AES, VDI, BIOS, XBIOS e.d.). Voor Multi-tasking en real-time toepassingen is er het module 'Process'.

## Ervaringen

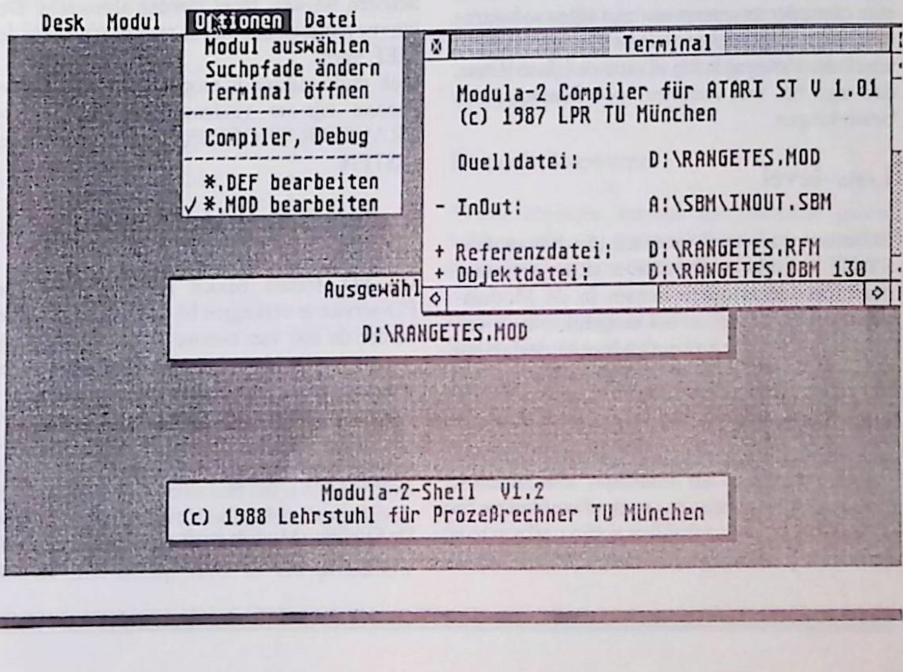
Tot nu toe is er nog niet veel ervaring met dit pakket, want het is pas onlangs ontvangen. Dus nog geen berichten over bugs, rare zaken en dergelijke. Wel horen wij graag over de ervaringen die men ermee heeft.

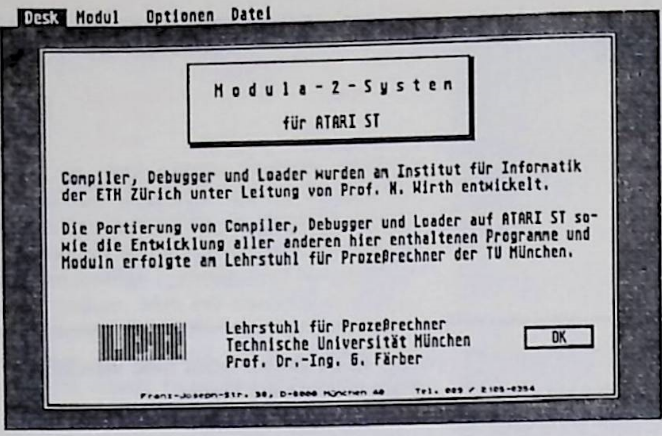
## Documentatie

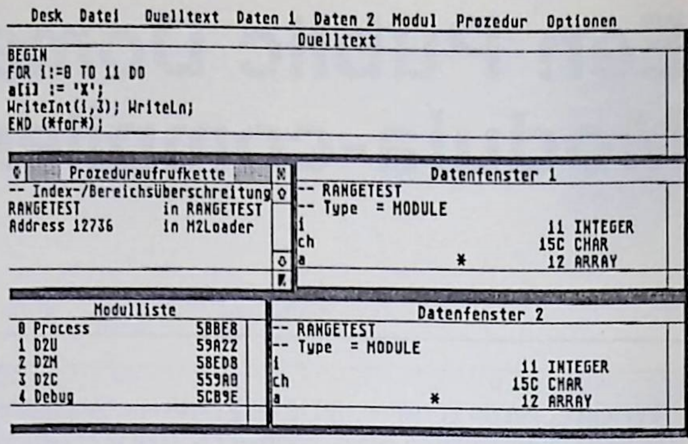
Er is niet erg veel documentatie bij het pakket. Maar een voorlopige verkenning heeft wel wat resultaten opgeleverd. Deze zijn op de disks erbij gezet.

## 16/32-bit problemen

Aangezien dit PD-Modula niet speciaal voor de ST is ontwikkeld, (of beter gezegd voor de






68000 CPU) zijn er moeilijkheden te verwachten met 16- en 32-bit zaken.
Zo is LONGWORD niet te vinden, maar wel LONGCARD en LONGINT en ADDRESS, die 32 bits zijn net zoals de POINTER's. Ook blijken datastructuren niet groter dan 32 Kb te kunnen zijn. Dit geldt ook voor stukken machinecode; bij gebruik van deze Modula-implementatie moeten er dus geen erg grote modules en procedures worden gebruikt.

## De programmeercyclus

Van belang is het dat de programmeur de cyclus editen - compileren -linken (hoeft hier niet) - draaien - fout - debuggen - editen enz snel kan doorlopen. Als men gebruik maakt van een Ram-disk of van een harde schijf, dan is deze cyclus zeer snel. Vooral het feit dat er bij het uittesten niet gelinkt hoeft te worden, scheelt heel veel tijd. In geval van compileerfouten komt men automatisch terug in de editor.

## Floating Point

Voor zover is uitgetest zijn de mathematische functies niet betrouwbaar. Er komen nogal vaak zinloze under- en overflows tot stand. De bewerkingen op integers en cardinals zijn o.k., het zijn de floating-point-routines die hier en daar niet deugen en tamelijk langzaam zijn. Het blijft jammer dat ons operating system op de ST geen floating-point-routines bevat, waardoor nu elke compiler en interpreter zijn eigen softwarefuncties nodig heeft, met alle fouten vandien, want als er iets erg lastig is, en moeilijk te testen, dan zijn het wel routines voor floating point bewerkingen.

## Low-level

Er is een low-level interface. In het module SYSTEM blijkt de INLINE te zitten, waarmee machinecode is op te nemen in de Modula-sourcetekst. Verder is het mogelijk om in een definition module een procedure te declareren met CODE(..); Hiervan is het gevolg dat elke aanroep van zo'n procedure resulteert in het uitvoeren van de machinecode-instruktie die in CODE is opgegeven. Dit is zeer handig bij het maken van trap- en interrupt- afhandelaars. Eventueel opgegeven parameters worden op de stack gezet, die de gebruiker zelf na afloop van de 'aanroep' moet opschonen (met INLINE).

Er is een functie SHIFT, die in staat is om bitpatronen te schuiven. Door een positief aantal shifts op te geven wordt naar links geschoven, en met negatieve waarden wordt naar rechts geschoven.
Met behulp van SETREG en REG kunnen de CPU registers gebruikt worden. De nummering van de registers is dezelfde als bij TDI Modula, dus CPU-register D0 is nummer 0, A0 is nummer 8 enz. Voor zover bekend, moet men van de registers D1, A4 tot en met A7 afblijven, aangezien deze door Modula gebruikt worden.

## Uitwisselbaarheid

Er is gebleken dat de compiler erg veel lijkt op de compiler van het nieuwe SPC-Modula. De files .SBM, .OBM en .RFM (uitvoer van de compiler) zijn in eenzelfde syntaxis genoteerd als bij het SPC-Modula. De bibliotheken kunnen dus wederzijds gebruikt worden!

## Diversen

Voor het dynamisch geheugenbeheer is er het module 'Heap', dat (onder andere) de Allocate en de Deallocate bevat.
SET's worden in maximaal 16 bits bijgehouden. Dus SET OF [0..17] gaat niet.
Reals zijn mogelijk, ook in dubbele precisie. Of de bijbehorende mathematische functies naar behoren werken, moet worden afgewacht. De interne representatie van de reals gaat volgens de IEEE norm.
Ook multi-tasking is mogelijk. In het module Process zijn te vinden: TRANSFER, IO-TRANSFER, PROCESS, NEWPROCESS en LISTEN.

## Verkrijgbaarheid

Dit PD-Modula pakket is via de normale PD-service te verkrijgen bij de Stichting ST. Zie verder de lijst van nieuwe PD en de Service-pagina in dit blad. Men kan de beide disks bestellen door gebruik te maken van de bestelkaart in dit blad.

## Ondersteuning

De Stichting Modula heeft zich bereid verklaard om te fungeren als centrale verzamelplaats van opmerkingen, vragen, kritiek en verbeteringen. Verdere berichten over dit PD-Modula in de volgende uitgaven van dit blad.

**Ronald van der Kamp**

*Stichting Modula*

---

*(vervolg van pag. 60)*

bestaat dat een muisklik enkel doorgegeven wordt als er ook een muisbeweging plaats vindt.
De beschijving van de beperkingen van de compiler zijn in het handboek nergens te vinden. Zo moet men zelf uitproberen wat de toegelaten types zijn van een tellende loop-variabele. Mij is gebleken dat LONGINT mag, maar LONGCARD niet. Maar mag CHAR of BYTE wel?
De compiler en linker geven beide beperkingen met de 32 Kbyte-limiet. De linker kan nog niet optimale code genereren. Het gebruik van de CPU-registers in de door de compiler gegenereerde code, kan beter.
De meeste van deze zwakheden zullen in de nabije toekomst worden verholpen.

## Conclusie

Al met al is dit een goed bruikbaar pakket. Op dit moment is het wachten op de volgende versie, die hopelijk vrij zal zijn van de meeste genoemde nadelen. Zeer hoopgevend is de goede support, die bij TDI op dit moment geheel is weggevallen. Evenwel is er voor Nederland nog geen importeur bekend, zodat men het zelf in Duitsland moet bestellen.

**Ronald J. van der Kamp**

## Naschrift:

In Duitsland is bekend geworden dat het SPC Modula het officiëele Atari Modula is geworden. Het pakket kan daar dus verkregen worden via de echte Atari dealers.

*SPC Modula-2*
*Producent:Advanced Applications Viczena GmbH Sperlingweg 19*
*D-7500 Karlsruhe 31*
*Beschikbaar gesteld door: Advanced Applications*
*Prijs DM 398,-*

# De Modula-Hoek

## De Scope-regels in Modula

**In de meeste leerboeken wordt er bijzonder weinig gezegd over de scope-regels. Het engelse woord 'scope' kan vertaald worden met 'bereik'. In deze aflevering zal ik het hebben over de regels die bepalen waar in een programmatekst welke zelfgedefinieerde namen gebruikt mogen worden. Immers de modulaire opbouw van programma's geeft ons de mogelijkheid om hiërarchische strukturen te maken en dan is, om vlot te kunnen programmeren, een goed begrip nodig van deze 'scope rules'.**

In het verdere verhaal zal ik de term 'object' af en toe gebruiken. Objekten zijn hetzij variabelen, hetzij type's, procedures of modules die van zelf verzonnen namen zijn voorzien. Een module bevat een aantal declaraties van objecten. Een module is een blok. Ook een procedure is een blok.
De positie van de declaratie van een objekt bepaalt waar het object bekend is (in welke blokken).

De eerste regel is:

"Naar een objekt kan verwezen worden in:
a. het blok waarin het is gedeclareerd,
b. in procedures die in datzelfde blok zijn gedeclareerd."

Het fundamentele idee hierbij is, dat alle variabelen, constanten, types en procedures die in een module worden gedeclareerd, in het gehele module bekend zijn en gebruikt kunnen worden. Dergelijke objekten worden dan als 'global' aangeduid. Het een en ander is aanschouwelijk gemaakt in listing 1.

```
MODULE Objekts;          0
VAR                      0
  tel: CARDINAL;         0
                         0
PROCEDURE DoeIets;       0
  VAR                    0 1
    tell: CARDINAL;      0 1
BEGIN (*DoeIets*)        0 1
.......                  0 1
END DoeIets;             0
                         0
BEGIN (*Objekts*)        0
.......                  0
END Objekts.             0
```

**LISTING 1.**

In dit voorbeeld is de variabele 'tel' bekend in het gehele module 'Objekts' en dus ook binnen de procedure 'DoeIets', maar 'tell' kan slechts gebruikt worden binnen de procedure 'DoeIets'.

Dit is in de listing aangegeven met de getalletjes rechts (de 0 en de 1).
Nu is het mogelijk om procedures binnen procedures te declareren. In listing 2 is dit

```
MODULE BuitenBlok;         0
FROM InOut IMPORT          0
     WriteLn,              0
     WriteCard;            0
                           0
                           0
VAR tel0: CARDINAL;        0
                           0
 PROCEDURE BuitenProc;     0
                           0 1
 VAR tell: CARDINAL;       0 1
                           0 1
 PROCEDURE BinnenProc;     0 1
                           0 1 2
  VAR tel2: CARDINAL;      0 1 2
                           0 1 2
  BEGIN (*BinnenProc*)     0 1 2
  tel2 := 5;               0 1 2
  tell := 6;               0 1 2
  WriteCard(tel3,4);       0 1 2
                           0 1 2
 END BinnenProc;           0 1
                           0 1
 BEGIN (* BuitenProc*)     0 1
  tel0 := 3;               0 1
  tell := 4;               0 1
  WriteLn;                 0 1
  BinnenProc;              0 1
                           0 1
 END    BuitenProc;        0
                           0
BEGIN (*BuitenBlok*)       0
                           0
 tell := 1;                0
 BuitenProc;               0
                           0
END BuitenBlok.
```

**LISTING 2.**

grafisch wat verduidelijkt door een blokstructuur in te tekenen. Hier ziet men een blok in een blok in een blok. Het Engelse woord voor dit effekt is 'nesting', en elke keer dat je weer in blok verder gaat, neemt het 'nesting- level' toe. Ik heb dit aangegeven met de cijfertjes rechts. Het module zelf zit op level 0. De procedure 'BuitenProc' maakt een nieuw blok op level 1, en 'BinnenProc' brengt het block-level op 2.
Let hier goed op de afgrenzingen van de blokken. Zo behoort de declaratie van de procedurenaam tot het blok dat er omheen zit; immers 'BuitenProc' kan overal in het module 'BuitenBlok' worden aangeroepen! Ook binnen het blok van 'BuitenProc' zelf kan 'BuitenProc' worden aangeroepen. Dit soort aanroepen heten dan 'recursief'. Verder kan gezegd worden dat 'BinnenProc' niet kan worden aangeroepen in de statements van het moduleblok, want 'BinnenProc' is niet binnen het bereik van het hoofdmodule gedeclareerd. BinnenProc' kan enkel worden gebruikt binnen het procedureblok van 'BuitenProc'. Er wordt gezegd dat BinnenProc 'local' is ten opzichte van Buitenproc.

De **tweede** Scope-regel:

"Objekten kunnen in een module binnengehaald worden door middel van een importlijst. Het bereik van de geïmporteerde objekten is het blok van het module."

In listing 2 is het te zien dat 'WriteCard' en 'WriteLn' het module binnen gebracht worden waarbij het verder lijkt alsof zij binnen het module blok zijn gedeclareerd. De zogenaamde standaard procedures zoals daar zijn INC, DEC, CAP en de standaard types als INTEGER en CHAR worden als het ware automatisch geïmporteerd in het module. Het gevolg van de tweede regel is dat deze namen 'gereserveerd' zijn; je kan ze niet zelf declareren als iets anders. Zou men bijvoorbeeld zowel WriteCard nodig hebben uit InOut als WriteCard uit het module Streams, dan moet men voortdurend duidelijk schrijven welke van de twee men bedoelt, dus InOut.Writecard of Streams.WriteCard. In listing 3 staat een voorbeeld van goede en foute constructies.

De **derde** Scope-regel:

"Twee objekten kunnen niet dezelfde naam hebben indien zij zijn gedeclareerd op eenzelfde block-level en binnenhetzelfde blok."

In listing 3 staat een voorbeeld met enkele fouten. Zo mag de procedure 'Tel' niet worden gedeclareerd, want die naam is al eerder in gebruik genomen in hetzelfde blok.
Binnen het blok van de procedure is men redelijk vrij in het bedenken van namen. Men kan binnen een procedure een objekt declareren met een naam die al in gebruik is in het blok dat de procedure omringt. Het objekt met die naam dat buiten de procedure in gebruik is, kan dan

```
Module DubbelOp;

CONST stopper=0;
      tel=4;

VAR tel: CARDINAL;
(* Fout: deze naam is *)
(* al in gebruik!!!!! *)

CONST stopper=10;
(* Fout: al eerder in *)
(* dit blok gebruikt! *)

tel1=30;
(* goed: nu een nieuwe*)
(* naam in gebruik    *)

PROCEDURE tel;
(* Fout: deze naam is *)
(* al in gebruik in   *)
(* dit blok.          *)

VAR muis:CARDINAL;
(* O.K. een nieuwe    *)
(* variabele in een   *)
(* nieuw blok.        *)

CONST muis=7;
(* Fout: in dit blok  *)
(* al in gebruik!     *)

BEGIN (*tel*)
.....
END tel;

BEGIN (*DubbelOp*)
.....
END DubbelOp.
```

**LISTING 3.**

niet worden bereikt en heeft dan een gat in zijn bereik. In listing 4 is dit te zien: de variabelen 'noppes' en 'tel' in het moduleblok kunnen niet gebruikt worden binnen de procedure 'Pasop' want daar worden de namen opnieuw gebruikt.

```
MODULE GatInScope;
CONST noppes = 0;
      tel = 1;
(* nu is dit geldig en  *)
(* verder is PasOp hier *)
(* ook bekend.          *)
VAR
....
PROCEDURE PasOp;
(* nu is er een andere  *)
(* noppes met de waarde *)
(* 7 geldig.Verder is de*)
(* tel nu een variabele *)
(* van het type CARDINAL*)
CONST noppes = 7;
VAR tel:CARDINAL;

BEGIN (*PasOp*)
.....
END PasOp;
(* noppes is nu een     *)
(* constante met de     *)
(* waarde 0, en tel     *)
(* is nu weer 1.        *)
(* Verder is hier ook   *)
(* PasOp bekend.        *)

BEGIN (*GatInScope*)
.....
END GatInScope.
```

**LISTING 4.**

Ik moet het hergebruik van namen ten sterkste afraden. Voor de compiler is het geen probleem, maar voor ons kan het lezen van programmateksten zeer lastig worden. Denk maar eens een programma in waarin alle tellers de letters I, J of K hebben, die in vele blokken worden gedeclareerd. En waarom zouden er zonodig gaten moeten vallen in het bereik van objekten?
De scope van een objekt wordt bepaald door de positie van de declaratie en niet door de plaatsen waar het objekt gebruikt wordt. Dit moet men voor ogen houden bij het aanroepen van procedures uit verschillende plaatsen in een programma. Zo zal het programma in listing 5 ervoor zorgen dat de waarde 20 wordt weggeschreven. Dit is zo, omdat 'UseNop' de waarde van de globale variabele 'noppes' gebruikt.

```
MODULE LetEensOp;
FROM InOut IMPORT
     WriteCard;
VAR noppes: CARDINAL;
PROCEDURE UseNop;
BEGIN
  WriteCard(noppes,5);
END UseNop;
PROCEDURE UseWat;
VAR noppes: CARDINAL;
BEGIN
noppes := 30;
UseNop;
END UseWat;

BEGIN (*LetEensOp*)
noppes := 20;
UseWat;
END LetEensOp.
```

**LISTING 5.**

De reden dat er 20 op het scherm komt te staan en niet 30 is als volgt te vatten:
'UseNop' schrijft de waarde van de variabele 'noppes' die gedeclareerd staat in het module 'LetEensOp'. Het feit dat de aanroep gedaan wordt in de procedure 'UseWat' waarin 'noppes' lokaal is gedeclarerd, doet er niet toe. 'UseNop' zou gedeclareerd moeten worden binnen 'UseWat' om de waarde 30 op het scherm te krijgen.
Natuurlijk zal een programmeur dit snel inzien, maar meestal ben je met je gedachten bij andere zaken dan dit soort trivialiteiten. Maar juist dit soort programmastrukturen kunnen het zeer lastig maken om een onbedoeld resultaat te verklaren.

De formele parameters van een procedure gedragen zich op dezelfde manier als andere variabelen. De namen van formele parameters (die in de declaratie beschreven worden) zijn lokaal ten opzichte van de procedure, dus het is net of ze een deel zijn van de proceduredeclaraties.

```
MODULE FormParNamen;
VAR tel1,
    tel2 : CARDINAL;
PROCEDURE NouJa
    (tel1: CARDINAL;
 VAR tel2: CARDINAL);
(*binnen deze proc.   *)
(* verwijzen tel1 en  *)
(* tel2 naar de para-*)
(*meters en niet      *)
(*naar de globale     *)
(*variabelen.         *)
BEGIN
tel1 := 1;
tel2 := 2;
END NouJa;

BEGIN
tel1 := 3;
tel2 := 4;
NouJa(tel1,tel2);
(* tel1 heeft nu de   *)
(*waarde 4, die was   *)
(*doorgegeven, maar   *)
(*tel2 heeft nu een   *)
(*andere waarde (in   *)
(* NouJa) gekregen    *)
(*namelijk 2, omdat   *)
(*deze variabele werd*)
(*doorgegeven als VAR*)
(* parameter.         *)
END FormParNamen.
```

**LISTING 6.**

Listing 7 toont een voorbeeld met goede verwijzingen naar namen. Binnen de procedure 'Bruikbaar' mag verwezen worden naar 'komtnog', verderop en ook naar 'InGebruik' omdat deze verwijzingen gebeuren in zijn opdrachten.
Vergeet dus niet, wanneer je iets declareert, even te kijken of de andere objekten die in de declaratie gebruikt worden al eerder in de tekst zijn gedeclareerd.

```
Module TeVoren

Procedure Bruikbaar;
BEGIN
komtnog := 10;
verderop := InGebruik()
END Bruikbaar;

VAR verderop,
    komtnog : CARDINAL;

PROCEDURE InGebruik():
          CARDINAL;
BEGIN
RETURN 59
END Ingebruik;

BEGIN
Bruikbaar
END TeVoren;
```

**LISTING 7.**

Er is nog een verdere regel met betrekking tot verwijzingen naar namen in declaraties.

De vierde Scope-regel:

"De namen die bedacht zijn voor constanten en types kunnen alleen dan gebruikt worden in andere declaraties als die declaraties later in de tekst verschijnen. Evenwel kunnen ze wel 'van te voren' gebruikt worden in opdrachten (statements)."

## De levensduur

Tot nu toe heb ik het gehad over het bereik van de naam van een objekt, en de regels die vastleggen waar in een programma een gedeclareerd objekt gebruikt kan worden. Van belang voor de scope-rules is de programmatekst, zoals de programmeur die op het scherm (of op papier) ziet. Dit wordt ook wel aangeduid als de 'statische' afbeelding. Maar vroeg of laat zal het

programma in werking gezet worden, en dan is de 'dynamische' kant van belang.
De levensduur waar het hier over gaat heeft betrekking op de momenten in de tijd dat objekten bestaan wanneer het programma uitgevoerd wordt. Dus wanneer objekten in de computer aanwezig zijn, en wanneer ze weer verdwijnen.
De regels voor de levensduur van objekten zijn:
1. een objekt dat gedeclareerd is op moduleniveau blijft bestaan zolang als het module aanwezig is.
2. een objekt dat lokaal in een procedure is gedeclareerd bestaat alleen zolang de procedure uitgevoerd wordt. Het objekt wordt aangemaakt zodra de procedure wordt aangeroepen, en verdwijnt na afloop van de uitvoering van de procedure. Wordt een procedure opnieuw aangeroepen, dan wordt er een nieuw objekt gemaakt met dezelfde naam als voorheen, en blijft bestaan totdat de procedure beëindigd wordt.
Een gevolg hiervan is dat lokaal gedefinieerde objekten hun waarden niet kunnen behouden tussen de aanroepen van de procedure.
Het idee van levensduur is dus verbonden met de inhoud van objekten, dus meestal met de waarde van variabelen.
In listing 8 staat een voorbeeld van een incorrect programma.

```
MODULE Levensduur;
FROM InOut IMPORT
            WriteCard;
VAR
tel: CARDINAL;

PROCEDURE DoeWat;
VAR tel1 : CARDINAL;
BEGIN
IF tel=1 THEN
  tel1 := 7
END; (*if*)
WriteCard(tel1,4);
END DoeWat;

BEGIN
tel := 1;
DoeWat;
tel := 2;
DoeWat
END Levensduur.
```

**LISTING 8.**

Ga maar na wat er daar gebeurt:
1. tel wordt op 1 gezet.
2. DoeWat wordt aangeroepen. Tel1 wordt gemaakt.
3. De If test ziet dat tel=1 en vult tel1 met de waarde 7.
4. de waarde 7 komt op het scherm.
5. DoeWat is klaar en de controle gaat terug naar het hoofd module. Hierbij verdwijnt tel1.
6. tel wordt op 2 gezet.
7. DoeWat wordt aangeroepen en een nieuwe tel1 wordt gemaakt. Let wel: tel1 wordt gemaakt, maar is nog niet met een waarde gevuld!
8. Tel=2 dus de IF opdracht heeft geen effekt.
9. Er wordt geprobeerd de inhoud van tel1 op het scherm te zetten

De reaktie van de computer kan uiteenlopen afhankelijk van de gebruikte compiler, TOS-versie en stand van zon en maan. Het kan zijn dat het getal 20 verschijnt op het scherm (puur toeval!). Maar ook een rare getalswaarde die nergens op slaat, of 23 bommen over het scherm, of een Modula run-time error, of een bevroren systeem behoren tot de mogelijkheden. Probeer het maar eens uit!

## Misbruik van globale variabelen

Er zijn twee manieren waarop in een procedure gebruik kan worden gemaakt van variabelen van buiten de procedure. De eerste methode is door gebruik te maken van parameters, en de tweede is gebruik te maken van variabelen die in hun bereik ook de procedure bevatten.
Het is zeer verwarrend als men beide methodes door elkaar gebruikt. Het effekt is namelijk dat een variabele binnen een procedure twee namen heeft, namelijk de naam van de declaratie in de procedure en de naam die voor hem gebruikt wordt als binnenkomende parameter, die, om het nog lastiger te maken, ook nog eens als VAR-parameter kan zijn opgegeven. In listing 9 staat een voorbeeld van zo'n verkeerd gebruik.

```
MODULE Var2Namen;
FROM InOut IMPORT
         WriteCard;
VAR temp,
    num:CARDINAL;

PROCEDURE Wissel
      (VAR temp1,
      temp2:CARDINAL);
(*verwissel inhouden*)
BEGIN
temp := temp1;
temp1:=temp2;
temp2:=temp
END Wissel;

BEGIN
temp:=3;
num:=4;
Wissel(temp,num);
WriteCard(temp,5);
WriteCard(num,5)
END Var2Namen.
```

**LISTING 9**

Bij het uitvoeren van het programma Var2Namen zal de inhoud van temp en num op het scherm verschijnen. Beide zijn 4. En dat is niet de bedoeling van het programma. De miskleun zit hierin dat de variabele 'temp' op module-niveau is gedeclareerd (hij is 'global') en mag als VAR-parameter worden doorgegeven. Wanneer Wissel wordt aangeroepen met als eerste parameter de variabele 'temp', dan zijn er wel twee namen in de procedure, namelijk 'temp' en 'temp1', maar beide verwijzen naar dezelfde variabele. Zodoende worden er als het ware drie opdrachten uitgevoerd:

temp := temp;
temp:= temp2;
temp2:=temp;

en zo zijn de inhouden van temp1 en temp2 niet verwisseld. De goede oplossing is gegeven in het programma van listing 10.

```
MODULE WelGoed;
FROM InOut IMPORT
          WriteCard;
VAR temp,
    num:CARDINEL;

PROCEDURE Wissel
      (VAR temp1,
      temp2:CARDINAL);
(*verwissel inhouden*)
VAR temp :CARDINAL;
BEGIN
temp := temp1;
temp1:=temp2;
temp2:=temp
END Wissel;

BEGIN
temp:=3;
num:=4;
Wissel(temp,num);
WriteCard(temp,5);
WriteCard(num,5)
END WelGoed.
```

**LISTING 10**

## Het gebruik van lokale en globale variabelen

Om een goed gebruik van de scope-regels te maken moet men zelf oefenen en wat programma's schrijven. Als vuistregel moet men aanhouden dat zaken pas daar gedeclareerd worden waar ze nodig zijn. Dus zo lokaal mogelijk. Slechts die stukken van het programma die het objekt nodig hebben, hoeven in zijn bereik te liggen.
1. als een variabele alleen in een procedure nodig is, declareer hem dan ook in die procedure. In een enkel, zeer zeldzaam, geval kan van deze regel worden afgeweken, namelijk als de verwerkingssnelheid van de computer ontoereikend is. De overweging is dan dat het aanmaken en weer doen verdwijnen van variabelen in een procedure (CPU-)tijd kost.
2. geef variabelen en waarden als parameters door aan procedures, behalve als je een goede reden hebt om het anders te doen.
Bijvoorbeeld: men maakt een module, met daarin een aantal ingewikkelde datastructuren met bijbehorende variabelen. In het module zijn er verder een aantal procedures, die alle deze variabelen gebruiken. Zouden die procedures worden geëxporteerd, dan hoeven de gebruikers van het module geen weet te hebben van de struktuur van de variabelen. De variabelen hoeven (en soms zou dat zelfs ongewenst zijn uit veiligheidsoverwegingen) niet geëxporteerd te worden. In een dergelijk geval is het zinnig dat de procedures direct naar de globale variabelen verwijzen in plaats van via parameterlijsten. Veelal merk je dit effect tijdens het schrijven van een module, doordat een aanroep van een procedure altijd dezelfde variabelen als parameters heeft.

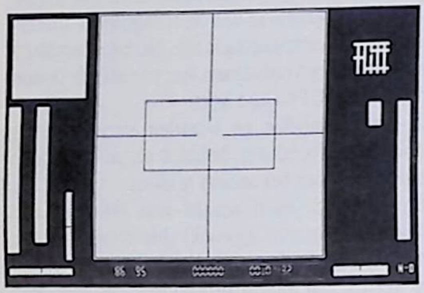
## Samenvatting

1. Objekten gedeclareerd op module-niveau zijn overal in het module toegankelijk (met
*(vervolg op pag. 68)*

# Public-domain software

**STICHTING ST Catalogus-Disk:**
**De complete Stichting ST disketten-bibliotheek nu op diskette. Deze diskette zal bij elke uitgave van een ST-nummer worden vervangen door de nieuwste lijst met Public Domain software. Deze Public Domain catalogus wordt geleverd met een demo-versie van het Nederlandse catalogiseerprogramma PROCAT. PROCAT is een schitterende disk-database met een prettige grafische weergave, uitgebreide zoek-, sorteer- en uitvoermogelijkheden.**
**STICHTING ST Catalogus:**
**In de uitgave nummer 16 van ons blad (november/december 1988) is een complete PD catalogus van 32 bladzijden opgenomen. Losse nummers van deze uitgave zijn via ons secretariaat verkrijgbaar.**

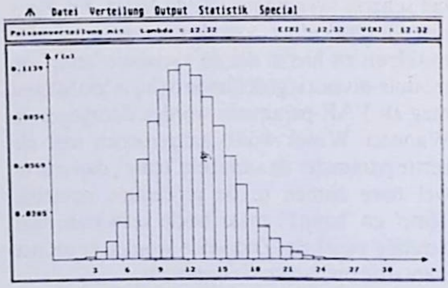
**Disk A 155** (DS)
LEERLINGENADMINISTRATIE: Dit programma is bedoeld om de leerlingenadministratie van een basisschool te verzorgen. Het programma is afgestemd op de gegevens die op de school van de maker worden gebruikt, waar dit programma al geruime tijd in gebruik is. Desgewenst kan het programma aan de wensen van andere gebruikers worden aangepast. Op diskette staan een kleuren- en een mono-versie.



*A 156 OPUS_1*

**Disk A 156** (SS)
OPUS-1: Aktie-spel met drie-dimensionale grafiek. In een ruim opgezet gangenstelsel moet u een vijandig vliegend object zien op te sporen en vernietigen. Door middel van de joystick stuurt u uw eigen sonde door het labyrint. Loopings en snelle wendingen zijn eenvoudig uit te voeren, maar pas op, want uw tegenstander kan in de volgende gang opduiken.

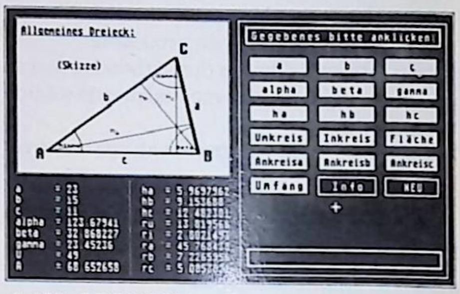
**Disk A 157** (SS)
FILEBOX: Vervanger van de systeem fileselector, die maar 7Kb geheugen nodig heeft. Loopwerken kunnen worden geselecteerd via het drukken van een button. Maximaal 16 loopwerken.
CMDSVR: Maakt het mogelijk een programma te starten, waarvan u niet weet waar het staat. Er wordt eenmalig een zoek-'path'-lijst aangemaakt, waarin de mogelijke 'paths' staan.
REMBO: Verwijdert alle REM's (opmerkingen in een Basic-regel) in een GFA-BASIC-programma, waardoor het programma zeer veel korter kan worden.
GOODVIEW: Loep, die een deel van het beeldscherm vergroot met een vergrotingsfactor van 2, 4, 8 of 16. De vergroting wordt geactiveerd door het drukken van de toetsen ALT-SHIFT-SHIFT. Werkt zowel in GEM- als in TOS-programma's.
CONVERT: Conversie-programma tussen de volgende verschillende grafische formaten: DEGAS, STAD, PWORKS, DOODLE, IMG, ART DIRECTOR, NEOCHROME. Dit programma houdt rekening met de resolutie en gecomprimeerde formaten.
FINDFILE: File-zoek-programma voor harddisks met optimale snelheid. Wildcards mogelijk (b.v.: *.DOC).




*A 158 DISKRET*

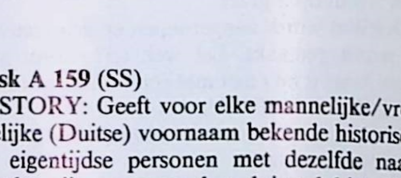
SETUP: Kies bij het op-booten welke programma's in de auto-folder en welke accessory's moeten worden gestart.
SUCHSETS: Laat de in een Signum-document gebruikte karaktersets zien, met het doel deze karaktersets naar dezelfde diskette of ramdisk te copiëren als waarop de documenten staan die van deze karaktersets gebruik maken.

**Disk A 158** (SS)
DISKRET: Grafische weergave van discrete verdeelfunkties; binomiaal, geometrisch of poisson. Als extra is de uitvoer naar een file mogelijk voor verdere bewerking.




*A 158 DRIEHOEK*

DREIECK: Het programma berekent uit een minimaal aantal gegevens van een driehoek, alle uit te rekenen waarden die bij die driehoek behoren.
PENDEL: Simulatie van een slinger met variabele demping en uitslag.
KÖRPER88: Berekening van geometrische lichamen. Dobbelsteen, prisma, (holle) cylinder, pyramide, kegel, pyramidestomp, kogel, kogelsnede, rechthoek, ruit, parallellogram, trapezium, driehoek, cirkel, ellips, enz..
KREDIT: Diverse berekeningen uit het kredietwezen.

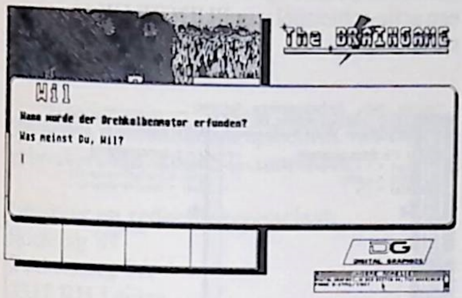



*A 158 KORPER88*

**Disk A 159** (SS)
HISTORY: Geeft voor elke mannelijke/vrouwelijke (Duitse) voornaam bekende historische en eigentijdse personen met dezelfde naam. Verder zijn er meer dan duizend historische gebeurtenissen en geboortedata van bekende persoonlijkheden opgeslagen, die na het invoeren van een bepaalde datum op het scherm verschijnen.

**Disk A 160** (SS)
BRAIN: Duitstalig quizspel voor veeleisenden. Er worden precies 1001 vragen gesteld uit verschillende wetenschapsgebieden. Het niveau is tamelijk hoog. Zeer aan te bevelen voor liefhebbers van 'Trivial Pursuit'.

**Disk A 161** (SS)
TRANSLATOR: Vertaalprogramma. Het programma geeft van een Duits woord de Engelse vertaling. De bibliotheek omvat ongeveer 8000 woorden.



*A 160 BRAIN*

**Disk A 162** (SS)
QUICKLEARN: Duitstalig vocabulaire programma met omvangrijke woordenverzameling. Dit woorden'boek' bevat een lijst met Engelse woorden en een lijst met vreemde woorden.
SCHEIBMASCHI[illegible]: Een vrij instelbare typecursus. Er kunnen [illegible]richte oefeningen worden gedaan, zowel vin[illegible] als tekstoefeningen. Het toetsenbord op het scherm is Duits, maar dit heeft geen invloed [illegible] uw typewerk.



*A 161 TRANSLATOR*

**Disk A 163** (SS)
H-WANDEL: Zet resource-files, die gemaakt zijn met de RSC (Resource Constuction Set) om in een GFA 3.0-source.
SWITCHER: Multifunctie utilitie met de volgende mogelijkheden: 50/60Hz omschakeling, beep aan/uitzetten, scherm inverteren, scherm uitzetten na 3 minuten, softwarematig stoppen van de computer, koude of warme reset.
X32-CONVERT: Converteert een Musix32-file naar een, met de systeemroutine DOSOUND, uitvoerbare vorm. Hierdoor kunnen muziekstukken zonder problemen in BASIC, C, enz. worden ingebonden.
ACCLOAD, DTA, CHANGE-ACC: Drie programma's om bij het opstarten van de computer een keuze te maken uit de aanwezige accessories.
GREIFER: Handig zoekprogramma voor stukken tekst. Zoekt in een file of in een op te geven path, één of meer tekstpassages en schrijft deze, naar wens, in een file. Veel mogelijkheden.



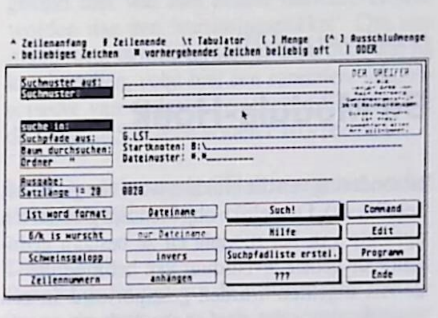
*A 162 QUICKLEARN*

**Disk A 164** (SS)
LABYRINTH: Behendigheidsspel. Door het bewegen van de muis wordt een tablet, met daarop een kogel, uit zijn horizontale neutrale positie bewogen waardoor de kogel gaat rollen. De speler moet proberen de kogel over het bord naar het einddoel te sturen. De opbouw van het bord kan worden veranderd en gesaved.



*A 162 SCHRIJFMASCHINE*

ROBOTER: Spel met veel levels, die door de speler bereikt kunnen worden door in elk level en sleutel te bemachtigen, waarmee een deur naar het volgende level kan worden geopend. U wordt echter steeds bedreigd door snelle robots en her en der geplaatste bommen.

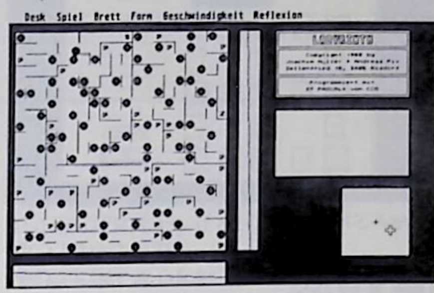


*A 163 GREIFER*

ZOMINO: Leuk uitgevoerd rekenprogramma, waarin de cijfers zijn vervangen door steeds andere symbolen. Probeer de oplossing te vinden en maak zelf opgaven. Variabelen maximaal vier cijfers lang.
SYMBOLIK: Hersenbreker als ZOMINO. Anders uitgevoerd en met variabelen van twee cijfers.

**Disk A 165** (SS,K)
SPEED MASTER: Arcade-aktie-spel met joystick voor mensen met een snelle reactie. In de vele levels op uw pad moet u voorwerpen verzamelen, waarbij u door allerlei tegenstanders zal worden gedwarsboomd. Vindt u echter een Powerpill, dan lust u deze tegenstanders rauw en heeft u daar geen last meer van.

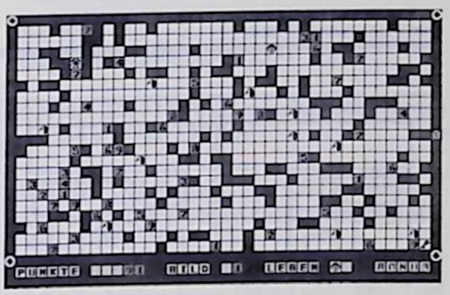
**Disk A 166** (SS)
FUNCALC: Programma voor numerieke rekenkunde. Invoer en berekening van variabelen naar keuze en functies, b.v. f(x) [- 2,3]. Directe vertaling in 68000-commando's zorgen voor een grote snelheid. Simpson-integraal berekening. Eigen desktop.



*A 164 LABYRINT*

PLOTTER: Handige kurve-plotter. Tot 5 functies, met elk maximaal 20 variabelen. Met o.a. Y-automaat, uitsnede vergroting, waardeberekening, raster, tekst invoeren, tekenfuncties. De gemaakte grafieken kunnen in andere programma's (tekstverwerker, tekenprogramma's) worden opgenomen.

**Disk A 167** (SS,K)
MEDAT: Kleur. Database voor een kleurenscherm. Het programma heeft vrij definieerbare maskers, waardoor het flexibel wordt voor allerlei toepassingen. Overzichtelijke bediening door het gebruik van kleur. Speciale mailmergefunctie voor 1st-Mail.
PROBASE: Mono. Professionele database net



*A 164 ROBOTER*

vrij definieerbare velden (tekst, integers, floating-point, datum, tijd, tekeningen). Verder mogelijkheden zoals: verplichte invoer en paswoord, comfortabele uitvoermogelijkheden, indexsequentieel gebruik.

**Disk A 168** (SS)
ERGOLOGIC: Edukatieve programma's voor kinderen van 5 tot 12 jaar. De programma's op deze diskette zijn speciaal geschreven als therapeutisch middel voor kinderen met lees- en schrijfmoeilijkheden, maar zijn ook goed bruikbaar voor kinderen met een normaal ontwikkelingsverloop. FLITS en SCHUD zijn leesoefeningen. KUNST, EENTWEE en DOOLHOF zijn grafische programma's, waarbij leesvaardigheid niet noodzakelijk is. Volledig in het Nederlands.

**Disk A 169** (DS,1 Megabyte)
MORIA: Dungeon simulatie-adventure voor één speler, die hier dagen, weken of maanden

SUPERCONTROL op HR-monitor f 39,- (f 45,-)
SUPERCONTROL op KL-monitor f 39,- (f 45,-)
SUPERCONTROL HR+KL f 70,- (f 80,-)

**Educatieve software**
ST-**Blokkendoos** programma disk, kleur of z/w f 45,-
ST-**Bouwdoos** editor en data disk in kleur f 85,-
**Bibliotheken:** Lezen 1, Rekenen 1,2 en 3, kleur of z/w f 19,50 per stuk
Disk **Puzzle** in kleur f 19,50
Totaal pakket in kleur f 179,50

De prijzen tussen haakjes gelden voor niet-abonnees.

**ProCat**
ProCat is een Nederlands programma dat het mogelijk maakt om een catalogus aan te leggen van al uw disks. Als u geregeld floppies en hard disk moet afzoeken naar bepaalde files (waar heb ik die file gelaten?), dan is dit het programma dat u nodig heeft. Een demo-disk is beschikbaar met onze volledige PD-catalogus erop (demo-disk f 10,-).

**ModPas**
Het is mogelijk om voor f 10,- via de Stichting ST de disk met Modulair Pascal te verkrijgen. Een bespreking is te vinden in ST nummer 16 van nov/dec 1988.

**PROCAT**
Momenteel is versie 1.5 klaar, die ook in kleur goed werkt.
De update kan verkregen worden door verzending van de originele disk met een cheque van f 10,- naar:
Atlantis Software
Postbus 60043
9703 BA Groningen
Er wordt gewerkt aan een versie die de mogelijkheid heeft om commentaar op te nemen in de catalogus.

Zie voor besprekingen van de programma's respectievelijk ST 8 (PRO- PRINT & GRACIA-FONT), ST 9 & 10 (ST- DIGGER), ST 13 & 14 (Midi PD). Voor ASDumper, zie de rubriek MIDI IN - MIDI OUT in ST 11 (derde jaargang nummer 1) en in ST 14.
De bespreking van de kleurenschermemulator en de zwart-wit emulator SUPERCONTROL kunt U vinden in ST 12 van maart/april 1988 (derde jaargang nummer 2)

**Bestelinformatie software**
Voor een vlotte toezending van de gewenste disks wordt U aangeraden om de bestelkaart in dit blad te gebruiken. Als U dan tegelijk een giro- of bankoverschrijving voor het juiste bedrag uitschrijft en beide gelijktijdig op de post doet, dan zal over het algemeen vijf dagen later dit bedrag op het gironummer 599626 van de Stichting ST, Bakkersteeg 9A, 2311 RH Leiden bijgestort worden. Verzending van de disks volgt binnen drie dagen na ontvangst van Uw betaling.
Let U ook op het juiste bedrag: de PD-disks kosten f 10,- per stuk.

**Verzendkosten**
Voor elke zending zijn wij genoodzaakt om een bedrag van f 2,50 aan verzendkosten in rekening te brengen.

## Hardware-connectoren

De ST-machines in hun diverse uitvoeringen hebben aan de achterkant aansluitingen met plugjes die soms moeilijk te krijgen zijn. Als service voor onze abonnee's heeft de Stichting ST besloten om een hardwareservice in stand te houden die al deze plugjes op korte termijn kan leveren.

De **monitorplug** met zijn 13 pinnen in een rechthoekig raster is zelden te vinden, alhoewel het ook een DIN-norm plug is. Deze plug kost momenteel f 5,-.

Een lastig te vinden plug is de grote ronde **floppyplug** met zijn 14 pinnen. Deze kost bij de Hardware- Service f 7,50.

Ook zijn de haakse printmontage floppy connectoren (vrouwtjes 14 pins) te verkrijgen voor f 5,50.

Voor mensen die een eigen Scart- kabel willen maken kan ook een **Scart-plug** geleverd worden (f 4,50). (zie ST nummer 4 van september 1986 voor een beschrijving van de Scart- aansluiting.)

De **DMA pluggen** (19 pin D-connectors) zijn ook moeilijk te krijgen. Momenteel zijn de snoerconnectoren (mannetjes) uit voorraad leverbaar. Ook is in beperkte mate de levering mogelijk van de contrapluggen. (printmontage, 19 pins, vrouwtjes, haaks) voor f 8,50 .

## Hardware-printservice

**Floppyconnector-print**
Het is mogelijk om een kaartje te leveren dat de doe-het-zelver in staat stelt om een eigen overgangsplug tussen de ronde 14-pins Atari floppyconnectorplug en de normale platte industriestandaard floppylintkabel, in elkaar te solderen (f 7,50).

**RS-232 monitorprint**
Het printplaatje dat behoort bij de RS-232 omschakelaar/tester beschreven in ST nummer 7 kost f 12,50 en is uit voorraad leverbaar.

## 7Updates

**ASDumper**
Van het programma ASdumper is een nieuwe update uitgekomen (versie 2.10).
Het aantal ondersteunde MIDI- apparaten is verder uitgebreid. Meer nieuws hierover staat in de MIDI-rubriek van uitgave nummer 14 (juli/augustus 1988). Geregistreerde gebruikers kunnen de software- bestelkaart in dit blad gebruiken om voor f 15,- een disk met nieuwe handleiding te bestellen o.v.v. 'update ASDUMPER' en het registratienummer van hun originele disk (groene stempel op de disk).

**Monitorschakelaar-print**
Het printplaatje dat behoort bij de in ST nummer 11 van januari 1988 besproken monitoromschakelaar kost f 10,-. De erbij benodige 13-pins haakse vrouwtjesprintmontage-pluggen zijn te verkrijgen voor f 4,50 per stuk en zijn uit voorraad leverbaar.
De complete set (minus schakelaar (2x om) en doosje) kost f 22,50.

**Floppy-bufferprint**
Momenteel is er een printje dat als buffer kan dienen tussen normale floppydrives (drieëneenhalf inch en vijfeneenkwart inch drives van andere merken dan Atari) en de ST. De ervaring heeft geleerd dat de signalen die door de sound chip gemaakt worden ter bediening van de drives, wat zwakjes van vermogen zijn. Dit simpele kaartje zal met behulp van een buffer chipje de signalen voldoende oppeppen.
Een bespreking van dit printplaatje is te vinden in ST nummer 12 en de prijs is f 10,-.

**Bestelinformatie hardware**
De onderdelen uit de Hardware-Service kunnen besteld worden door gebruik te maken van de bestelkaart in dit blad.

**Verzendkosten**
Voor elke zending zijn wij genoodzaakt om een bedrag van f 2,50 aan verzendkosten in rekening te brengen.

## Boeken

**Professionel GEM va[illegible]n Oren**
Eris nu de tweede, her[illegible] editie beschikbaar. De handleiding, die 17 [illegible]fdstukken bevat en 236 pagina's dik is, wor[illegible]leverd in een 3-rings A5-mapje met opdruk.

In ST van december 19[illegible] (nummer 5, blz. 20) en juni 1987 (numm[illegible] 7 blz. 22) is een beschrijving van de i[illegible]oud te vinden. De C-sources voor de hoofd[illegible]kken 2 t/m 16 zijn te vinden op de disk ST11, (de disk die bij ST 11 (januari 1988, 3e jaargang nr. 1) hoort).
Het ziet er naar uit dat er geen verdere aanvullingen van Tim Oren te verwachten zijn i.v.m. een verandering in zijn broodwinning.

Bestelling van de complete handleiding is mogelijk door het overschrijven van f 27,- op giro 1626826 van
de Stichting ST,
Bakkersteeg 9A
2311 RH Leiden
onder vermelding van 'Tim Oren Handleiding'.

BESTELKAART HARDWARE & SOFTWARE

**Ik bestel de volgende Software / Hardware:**

| Aantal | Omschrijving | Prijs | Totaal |
|---|---|---|---|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Handtekening ............................ Verzendkosten ............................

Bedrag overmaken op giro 599626
t.n.v. Stichting ST, Leiden.
Voor België: bankrek. 390-0956760-07
t.n.v. V.A.S.T. te Gent

Totaal ............................

Omrekening voor België: 20 bfrs = f.1,-

le post

a om post te

met ASCII-
len
ken van eti-

automatisch
- en verza-
en



/dataprogramma

werkt ook zonder Pull Down menu's

**Optimaal gebruik van uw printer:**

- That's Write ondersteunt zowel de grafische als de tekst modus bij 9 en 24 naalds printers (eventueel in een regel)
- 24 naalds printers in draft, 180 * 360 DPI en 360 * 360 DPI, Download fonts
- interne fonts van de printer, GEM-Fonts in elke grootte en ingebouwde vektorfonts van de printer worden ondersteund

**WYSIWYG:**

- regelafstand 1, 1.5, 2, 2.5 en 3 regels
- tekeningen kunnen verplaatst worden en groter en kleiner gemaakt worden
- ondersteunt monochroom, kleur en grootbeeld monitoren
- automatisch teksten bewaren na een zelf te bepalen tijd
- knippen en invoegen van blokken d.m.v. vier onafhankelijke buffers
- ASCII bewaren en laden, met keuze mogelijkheid CR/LF aan einde van elke regel of niet
- 10 merkpunten om snel een tekstpassage te zoeken, kunnen met de tekst mee bewaard worden
- printen via paralelle, serieële of DMA poort en d.m.v. file
- gratis hotline voor geregistreerde gebruikers
- mogelijkheid om met kolommen te werken, maximaal 5

**voor de ATARI-ST:**

- nieuw viewdata communicatie programma voor hoge resolutie monitor
- om op efficiente manier informatie in viewdatabanken te raadplegen, onderhouden en bewerken
- gemakkelijk te bedienen en overzichtelijk

## That's Stocks

**Kompleet aandelen programma:**

- bekijken, bereken, manipuleren en beoordelen van fondsen die genoteerd staan op de effectenbeurs of goederenmarkt
- effectenportefeuille bijhouden
- tabellen, overzichten, grafieken van een of meerdere fondsen
- makkelijk bedienbaar

*That's Write* - de internationale tekstverwerker voor de Atari ST.
*That's Write* - die internationale Textverarbeitung für den Atari ST.
*That's Write* - the international word processor for the Atari ST.
*That's Write* - le traitement de texte internationale pour l'Atari ST.
*That's Write* - la gestione dei testi internazionale per l'Atari ST.
*That's Write* - международная обработка текста для персонального компютера 'Atari ST'.
*That's Write* - الكاتب الدولي لاتاري س ت

Demodiskettes *That's Write* en *That's Stock*. Stort per programma fl. 10,00 op bankrekening 13.18.13.137 van **COMPO,** Postbus 20, 6269 ZG, Margraten.

That's Write

PRESENTED BY
COMPO

FOR ATARI ST

THAT'S write